



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS184853

**RANCANG BANGUN APLIKASI E-INKUBATOR:
PLATFORM PENYEDIA LAYANAN INKUBASI DAN
INVESTASI BAGI UMKM BERBASIS MICROSERVICE**

**DEVELOPMENT OF MICROSERVICE BASED
APPLICATION E-INKUBATOR: INCUBATION AND
INVESTATION SERVICE PROVIDER FOR UMKM**

WILDAN AZKA FILLAH
0521154000077

Dosen Pembimbing
Nisfu Asrul Sani S.Kom., M.Sc.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR - KS184853

**RANCANG BANGUN APLIKASI E-INKUBATOR:
PLATFORM PENYEDIA LAYANAN INKUBASI
DAN INVESTASI BAGI UMKM BERBASIS
MICROSERVICE**

WILDAN AZKA FILLAH
05211540000077

Dosen Pembimbing
Nisfu Asrul Sani S.Kom., M.Sc.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

UNDERGRADUATE THESIS - KS184853

**DEVELOPMENT OF MICROSERVICE BASED
APPLICATION E-INKUBATOR: INCUBATION
AND INVESTATION SERVICE PROVIDER FOR
UMKM**

WILDAN AZKA FILLAH
05211540000077

Supervisor
Nisfu Asrul Sani S.Kom., M.Sc.

INFORMATION SYSTEM DEPARTMENT
Information Technology and Communication Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI E-INKUBATOR:
PLATFORM PENYEDIA LAYANAN INKUBASI DAN
INVESTASI BAGI UMKM BERBASIS MICROSERVICE**

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

WILDAN AZKA FILLAH

NRP. 05211540000077

Surabaya, 25 Januari 2019

**KEPALA
DEPARTEMEN SISTEM INFORMASI**



Mahendrawathi ER, S.T., M.Sc., Ph.D

NIP. 19761011 200604 2 001

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI E-INKUBATOR: PLATFORM PENYEDIA LAYANAN INKUBASI DAN INVESTASI BAGI UMKM BERBASIS MICROSERVICE

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

WILDAN AZKA FILLAH

NRP. 05211540000077

Disetujui Tim Penguji : Tanggal Ujian : 21 Januari 2019
Periode Wisuda : Maret 2019

Nisfu Asrul Sam S.Kom., M.Sc.

Dr. Ir. Aris Tjahyanto M.Kom.

Hatma Suryotrisongko, S.Kom., M.Eng.

(Pembimbing I)

(Penguji I)

(Penguji II)



RANCANG BANGUN APLIKASI E-INKUBATOR: PLATFORM PENYEDIA LAYANAN INKUBASI DAN INVESTASI BAGI UMKM BERBASIS MICROSERVICE

Nama Mahasiswa : Wildan Azka Fillah
NRP : 0521154000077
Departemen : Sistem Informasi FTIK-ITS
Pembimbing : Nisfu Asrul Sani S.Kom, M.Sc.

ABSTRAK

Perkembangan zaman membuat banyak orang menggunakan teknologi informasi dalam dunia bisnis tidak terkecuali UMKM. Perangkat lunak yang digunakan UMKM salah satunya adalah E-inkubator, platform penyedia layanan inkubasi dan investasi. Perangkat lunak tersebut menggunakan arsitektur monolitik yang seiring bertambah banyaknya user membuat waktu respon dari aplikasi menurun. Aplikasi ini juga belum dapat diintegrasikan dengan sistem yang ada di pemerintah maupun perusahaan sebagai investor guna mempermudah proses investasi.

Masalah tersebut dapat diselesaikan dengan penerapan arsitektur microservice pada aplikasi E-inkubator. Konsep Microservice yang loosely coupled memudahkan pihak eksternal dalam mengakses service dari aplikasi ini. Konsep tersebut juga membuat respon aplikasi menjadi lebih cepat terhadap banyaknya permintaan dikarenakan service pada aplikasi yang terpisah-pisah.

Pada tugas Akhir ini dilakukan rancang bangun pengembangan aplikasi E-inkubator berbasis microservice dengan menggunakan metode Waterfall. Aplikasi ini diharapkan dapat mempermudah pihak eksternal seperti pemerintah, perusahaan, atau inkubator dalam mengakses service dari aplikasi ini dan mengintegrasikan dengan sistem mereka. Selain itu, aplikasi ini juga diharapkan dapat mempercepat waktu respon terhadap permintaan yang tinggi.

Kata Kunci : Investasi, Inkubasi, Microservice, UMKM

DEVELOPMENT OF MICROSERVICE BASED APPLICATION E-INKUBATOR: INCUBATION AND INVESTATION SERVICE PROVIDER FOR UMKM

Name : Wildan Azka Fillah
NRP : 0521154000077
Department : Information System FTIK-ITS
Supervisor : Nisfu Asrul Sani S.Kom, M.Sc.

ABSTRACT

This current development make people use information technology in business no exception for small and medium enterprises (SMEs). Softwares that used by SMEs are many, one of them is E-inkubator, incubation and investment service provider platform. This software still use monolithic architecture, but as users increase, make response time of that application decrease. This application also cant integrated with existing system in governments or enterprises as an investor in order to make easier investment process.

This problem can be solved by implement microservice architecture in this application. Microservice concept is loosely coupled. It make easier third party to access service from this application. Those concept also make response time from application faster toward many request. It happens because services in application is separated.

This final project will make design and develop E-Inkubator application that based on microservice architecture use Waterfall Development Lifecycle. This application expected to make easier third party like governments, enterprises, or others incubators to access services from this application and integrate to their systems. In addition, this application also expected make response time faster toward many request.

Keywords: *Investation, Incubation, Microservice, SME*

KATA PENGANTAR

Dengan mengucapkan rasa syukur kepada Tuhan Yang Maha Pengasih dan Maha Penyayang atas izin-Nya penulis dapat menyelesaikan buku yang sederhana ini dengan judul Rancang Bangun Aplikasi E-Inkubator: Platform Penyedia Layanan Inkubasi dan Investasi bagi UMKM berbasis Microservice. Dalam penyelesaian Tugas Akhir ini, penulis diiringi oleh pihak-pihak yang selalu memberi dukungan, saran, dan doa sehingga penelitian berlangsung dengan lancar. Secara khusus penulis mengucapkan terima kasih dari lubuk hati terdalam kepada:

1. Tuhan, yang selalu menemani dan membimbing penulis dalam segala aspek kehidupan.
2. Ibu Mahendrawathi ER, S.T., M.Sc., Ph.D selaku Ketua Departemen Sistem Informasi ITS Surabaya.
3. Bapak Nisfu Asrul Sani, S.Kom, M.Sc. selaku dosen pembimbing yang telah mencurahkan segenap tenaga, waktu dan pikiran dalam penelitian ini, serta memberikan motivasi yang membangun.
4. Bapak Dr. Ir. Aris Tjahyanto M.Kom. dan Bapak Hatma Suryotrisongko, S.Kom, M.Eng. selaku dosen penguji yang telah memberikan kritik dan saran yang membuat kualitas penelitian ini lebih baik lagi.
5. Segenap dosen dan karyawan Departemen Sistem Informasi.
6. Orang tua penulis, yang tiada hentinya mendoakan dan memberikan dukungan kepada penulis.
7. Segenap keluarga penulis yang telah mendoakan dan memberikan dukungan kepada penulis.
8. Dan pihak lainnya yang berkontribusi dalam tugas akhir yang belum dapat penulis sebutkan satu per satu.

Penyusunan tugas akhir ini masih jauh dari kata sempurna, untuk itu penulis menerima segala kritik dan saran yang membangun sebagai upaya menjadi lebih baik lagi ke depannya.

Semoga buku tugas akhir ini dapat memberikan manfaat untuk pembaca.

Surabaya, Januari 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERSETUJUAN.....	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Permasalahan.....	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Relevansi.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian Sebelumnya	5
2.2 Microservice	6
2.3 Laravel/Lumen	7
2.4 E-Inkubator	8
2.4.1 UMKM.....	9
2.4.2 Coach	9
2.4.3 Investor	10
2.4.4 Inkubator	10
2.5 <i>Waterfall</i> SDLC	10
2.6 Unified Modeling Language (UML).....	10
2.7 Blackbox Testing	11
2.8 Load Testing	11
2.9 Stress Testing.....	11
BAB III METODOLOGI.....	13
3.1 Tahapan Pelaksanaan Tugas Akhir.....	13
3.1.1 Studi Literatur	13
3.1.2 Penggalan Kebutuhan	13
3.1.3 Desain Sistem.....	13

3.1.4 Implementasi Desain Sistem	14
3.1.5 Testing.....	15
3.1.6 Deployment.....	15
3.1.7 Dokumentasi	15
3.2 Jadwal Kegiatan	15
BAB IV ANALISIS KEBUTUHAN DAN PERANCANGAN.....	17
4.1 Gambaran Umum.....	17
4.2 Analisis Kebutuhan Sistem	17
4.2.1 Analisis Kebutuhan Fungsional	17
Dalam aplikasi E-Inkubator, dibutuhkan fungsi seperti pada tabel 4.1.	17
4.2.2 Analisis Kebutuhan Microservice	18
4.3 Desain Sistem.....	21
4.3.1 Use Case Diagram.....	21
4.3.2 Use Case Description	21
4.3.3 Class Diagram	36
4.3.4 Component Diagram	36
4.3.5 Desain Database	38
BAB V IMPLEMENTASI	41
5.1 Lingkungan Implementasi.....	41
5.2 Implementasi Desain Aplikasi	41
5.2.1 Implementasi Melihat Investasi	41
5.2.2 Implementasi inkubasi service	43
5.2.3 Login	43
5.2.4 Melihat Profil	45
5.2.5 Memperbaharui profil	46
5.2.6 Melihat History Coaching	47
5.2.7 Melakukan Coaching	48
5.2.8 Melihat Investasi yang diterima	49
5.2.9 Melihat detail Investasi yang diterima	50
5.2.10 Menambahkan Investasi.....	51
5.2.11 Melihat Investasi yang telah dilakukan	52
5.2.12 Melihat List Umkm	53
5.2.13 Melihat detail Umkm	54
5.2.14 Melihat Ukm yang aktif coaching.....	55
5.3 Implementasi Test Case	56

5.4 Implementasi REST API pada perangkat lain	57
BAB VI HASIL DAN PEMBAHASAN	59
6.1 Arsitektur Microservice	59
6.2 Stress Testing.....	60
6.3 Penggunaan service pada Android.....	61
BAB VII KESIMPULAN DAN SARAN	63
7.1 Kesimpulan	63
7.2 Saran	63
LAMPIRAN A	67
BIODATA PENULIS	69

DAFTAR GAMBAR

Gambar 2.1 Diagram Arsitektur <i>microservice</i>	6
Gambar 2.2 Diagram Proses Bisnis Inkubasi	8
Gambar 2.3 Diagram Proses Bisnis Investasi.....	9
Gambar 3.1 Arsitektur Monolitik E-Inkubator.....	14
Gambar 3.2 Arsitektur Microservice E-Inkubator	14
Gambar 4.1 Use Case Diagram	21
Gambar 4.2 Class Diagram	36
Gambar 4.3 Component Diagram	37
Gambar 4.4 Detail Component Diagram.....	38
Gambar 4.5 Desain Database	39
Gambar 5.1 Front-end Melihat Investasi.....	42
Gambar 5.2 Potongan Kode Front-end Melihat Investasi.....	42
Gambar 5.3 Potongan Kode Microservice Melihat Investasi.....	42
Gambar 5.4 Tampilan Melakukan Inkubasi	43
Gambar 5.5 Potongan Kode Microservice Melakukan Inkubasi	43
Gambar 5.6 Front-end Login.....	44
Gambar 5.7 Potongan Kode Front-end Login	44
Gambar 5.8 Potongan Kode Microservice Login.....	45
Gambar 5.9 Tampilan Profil Diri	45
Gambar 5.10 Potongan Kode Menampilkan Profil	46
Gambar 5.11 Tampilan Front-end Memperbaharui Profil	46
Gambar 5.12 Potongan Kode Microservice Memperbaharui Profil.....	47
Gambar 5.13 Tampilan Front-end History Coaching.....	47
Gambar 5.14 Potongan Kode Front-end History Coaching ...	47
Gambar 5.15 Potongan Kode Microservice History Coaching	48
Gambar 5.16 Tampilan Coaching	48
Gambar 5.17 Potongan Kode Front-end Coaching	48
Gambar 5.18 Potongan Kode Microservice Coaching.....	49
Gambar 5.19 Tampilan Front-end Investasi yang diterima....	49
Gambar 5.20 Potongan Kode Front-end Investasi yang diterima	49
Gambar 5.21 Potongan Kode Microservice Investasi yang diterima	50

Gambar 5.22 Tampilan Front-end Detail Investasi	50
Gambar 5.23 Potongan Kode Front-end Detail Investasi.....	51
Gambar 5.24 Potongan Kode Microservice Detail Investasi ..	51
Gambar 5.25 Tampilan Front-end Menambahkan Investasi ..	51
Gambar 5.26 Potongan Kode Front-end Menambahkan Investasi.....	52
Gambar 5.27 Potongan Kode Microservice Menambahkan Investasi.....	52
Gambar 5.28 Tampilan Front-end Melihat Investasi.....	53
Gambar 5.29 Potongan Kode Front-end Melihat Investasi	53
Gambar 5.30 Potongan Kode Microservice Melihat Investasi	53
Gambar 5.31 Tampilan Front-end List Ukm	53
Gambar 5.32 Potongan Kode Front-end List Ukm	54
Gambar 5.33 Potongan Kode Microservice List Ukm	54
Gambar 5.34 Tampilan Front-end Detail Ukm	54
Gambar 5.35 Potongan Kode Front-end Detail Ukm.....	55
Gambar 5.36 Potongan Kode Microservice Detail Ukm.....	55
Gambar 5.37 Tampilan Front-end Ukm Aktif Coaching	55
Gambar 5.38 Potongan Kode Front-end Ukm Aktif Coaching	56
Gambar 5.39 Potongan Kode Microservice Ukm Aktif Coaching.....	56
Gambar 5.40 Penggunaan API pada Java Android	58
Gambar 5.41 Tampilan Hasil pada Android.....	58
Gambar 6.1 RPS Microservice	59
Gambar 6.2 RPS Monolitik	59
Gambar 6.3 Performa CPU dan RAM.....	61
Gambar 6.4 Kode Menghitung Response Time Service	62
Gambar 6.5 Hasil Uji Coba Response Time	62

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya	5
Tabel 3.1 Jadwal Kegiatan	16
Tabel 4.1 Kebutuhan Fungsional	17
Tabel 4.2 Kebutuhan Microservice UMKM	19
Tabel 4.3 Kebutuhan Microservice Coach.....	19
Tabel 4.4 Kebutuhan Microservice Investor	20
Tabel 4.5 Kebutuhan Microservice Inkubator.....	20
Tabel 4.6 Use Case Description Login.....	22
Tabel 4.7 Use Case Description Melihat Profil.....	23
Tabel 4.8 Use Case Description Memperbaharui Profil.....	23
Tabel 4.9 Use Case Description Berdiskusi dengan Coach ..	24
Tabel 4.10 Use Case Description Melihat Riwayat Coaching	25
Tabel 4.11 Menerima Investasi	26
Tabel 4.12 Menanggapi Pesan UKM	27
Tabel 4.13 Use Case Description Mencari Data Masalah dan Solusi.....	28
Tabel 4.14 Use Case Description Melakukan Investasi	29
Tabel 4.15 Use Case Description Melihat Histori Investasi ..	30
Tabel 4.16 Use Case Description Melihat List UKM	31
Tabel 4.17 Use Case Description Melakukan Verifikasi Investasi.....	31
Tabel 4.18 Use Case Description Melakukan Manajemen User	32
Tabel 4.19 Use Case Description Melakukan Manajemen Investasi.....	33
Tabel 4.20 Use Case Description Manajemen Coaching	34
Tabel 5.1 Spesifikasi Perangkat Keras	41
Tabel 5.2 Spesifikasi Perangkat Lunak	41
Tabel 7.3 Skenario Uji Coba	56
Tabel 8.1 Hasil Uji Coba Load Testing	60
Tabel 8.2 Hasil Uji Coba Stress Testing	60

BAB I

PENDAHULUAN

Pada bab ini berisi mengenai gambaran umum tugas akhir meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat tugas akhir, dan relevansi tugas akhir dengan Laboratorium Infratraktur dan Keamanan Teknologi Informasi.

1.1 Latar Belakang

Perkembangan teknologi pada zaman ini membuat banyak orang menggunakan teknologi informasi tidak terkecuali Usaha Mikro, Kecil, dan Menengah (UMKM). UMKM pada saat ini banyak menggunakan teknologi informasi seperti internet dalam menjalankan proses bisnis mereka. Sebagai contoh penggunaan TI seperti menerima pesanan atau mempromosikan usaha mereka. Salah satu perangkat lunak yang digunakan UMKM adalah E-inkubator, suatu aplikasi investasi dan inkubasi daring. Aplikasi E-inkubator pada umumnya bermanfaat bagi UMKM yang ingin mengembangkan bisnisnya dan mencari investasi.

Aplikasi E-inkubator yang sekarang merupakan suatu sistem yang menerapkan arsitektur monolitik. Arsitektur monolitik dibangun menggunakan pendekatan dimana seluruh aplikasi dibangun, dirancang, diimplementasikan, dan dipelihara dalam satu kesatuan kode [1]. Kelemahan arsitektur ini apabila aplikasi sudah berkembang menjadi besar, maka pengembang akan kesulitan dalam melakukan scaling. Pengembangan berkelanjutan juga akan sulit, karena ketika suatu komponen diperbaharui, maka seluruh komponen dalam kesatuan aplikasi juga harus diperbaharui [2].

Jumlah UMKM di Indonesia sangat banyak yakni sebesar 59.2 juta [3]. Jumlah tersebut membuat permintaan/*request* dari pengguna terhadap layanan dalam aplikasi E-inkubator juga tinggi. Sistem monolitik yang sekarang diterapkan dapat

membuat aplikasi turun performa responsenya sehingga membuat aplikasi menjadi lambat.

Pada awal tahun 2011, muncul istilah baru dalam dunia *Software Architect*, yaitu *microservice* [4]. Secara sederhana hal ini merupakan *software* atau sistem informasi dirancang untuk terdistribusi dan memberikan layanan spesifik dan terfokus. Pentingnya *microservice* dalam pengembangan sebuah aplikasi adalah untuk mempercepat permintaan/*request* terhadap layanan pada aplikasi karena *service* terpisah atau *loosely coupled*. *Microservice* juga membuat aplikasi lebih dapat diatur dan *scalable* [5].

Penggunaan teknologi *microservice* dalam penerapan pada aplikasi E-inkubator dapat memberikan beberapa kelebihan. Konsep *microservice* memberikan beberapa manfaat. Apabila pengembang ingin melakukan pengembangan yang melibatkan satu atau lebih *service* tertentu maka *service* lain tidak akan terganggu. Selain itu pengembangan suatu *service* dapat dikembangkan menggunakan bahasa pemrograman yang berbeda-beda [6].

Penerapan *microservice* pada Aplikasi E-inkubator dengan konsepnya yang *loosely coupled* menjamin kecepatan dalam pemberian layanan, kecepatan penanganan masalah, dan memudahkan pengembang dalam menyesuaikan permintaan *capacity* dari pengguna terhadap suatu *service* [7]. Konsep *loosely coupled* dalam *microservice* juga dapat dimanfaatkan agar *service* yang terpisah tersebut digunakan oleh pengguna lain seperti pemerintah dengan mengintegrasikannya ke sistemnya.

Pengembangan aplikasi *microservice* “E-inkubator” akan menggunakan modifikasi/turunan dari framework Laravel yang bernama Lumen. Penggunaan framework memberikan beberapa keuntungan khususnya bagi pemula dalam menggunakan framework ini. Lumen juga mendukung pengembangan *microservice* dengan mendukung teknologi RESTful API sebagai salah satu komponen penting dari *microservice* untuk pertukaran data.

Pengerjaan tugas akhir ini, diharapkan dapat mengembangkan aplikasi E-inkubator berbasis microservice agar dapat diambil manfaat dari teknologi tersebut yakni dengan meningkatkan kecepatan dalam pelayanan layanan/*service* dan pemanfaatan *loosely coupled* dari *service* pada aplikasi ini.

1.2 Rumusan Masalah

Rumusan Masalah dari penelitian ini adalah:

1. Bagaimana arsitektur aplikasi E-inkubator menggunakan arsitektur microservice?
2. Bagaimana merancang dan membangun aplikasi E-inkubator dengan arsitektur microservice?

1.3 Batasan Permasalahan

Batasan masalah dalam tugas akhir ini adalah:

1. Aplikasi dikembangkan pada framework Laravel/Lumen.
2. Aplikasi di deploy pada server lokal.

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk bisa mengembangkan aplikasi E-inkubator: platform penyedia layanan inkubasi dan investasi berbasis microservice.

1.5 Manfaat

Manfaat penelitian pada tugas akhir ini adalah dapat mengembangkan aplikasi E-inkubator: platform penyedia layanan inkubasi dan investasi berbasis microservice agar dapat dimanfaatkan oleh penggunanya.

1.6 Relevansi

Tugas Akhir ini memiliki luaran berupa implementasi arsitektur microservice pada aplikasi. Sehingga relevansi tugas akhir ini terhadap laboratorium IKTI adalah dengan penerapan arsitektur microservice guna mendukung isu strategis yang sekarang menjadi fokus penelitian pada laboratorium IKTI tahun 2015-

2020 yakni Desain Arsitektur Software: Resilient Information System. Software/ Sistem informasi termasuk kategori resilient apabila software/sistem informasi tersebut memiliki kemampuan dalam menyesuaikan kebutuhan baru yang belum ada pada desain sebelumnya. Dengan penerapan arsitektur microservice ini, setiap kebutuhan didesain menjadi service, sehingga apabila ada kebutuhan bisnis yang bertambah tanpa merubah service lain.

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka terdiri dari landasan-landasan yang akan digunakan dalam penelitian tugas akhir ini, mencakup penelitian-penelitian sebelumnya, kajian pustaka, dan metode yang digunakan selama pengerjaan.

2.1 Penelitian Sebelumnya

Terdapat beberapa penelitian yang memiliki topik yang hampir serupa dengan penelitian ini, diantaranya akan dijelaskan pada Tabel 2.1

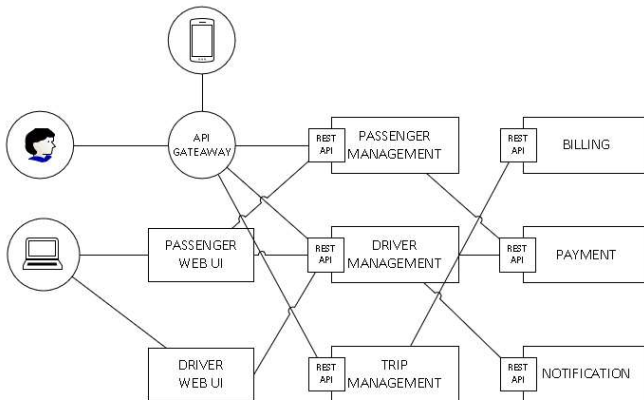
Tabel 2.1 Penelitian Sebelumnya

Nama Peneliti	Rama Rahmanda [8]
Judul Penelitian	Rancang Bangun Aplikasi berbasis Microservice untuk Klasifikasi Sentimen. Studi Kasus: PT. Yesboss Group Indonesia (Kata.ai)
Deskripsi	Pada penelitian ini ditemukan bahwa layanan chatbot pada perusahaan Kata.ai memiliki tingkat penggunaan yang tinggi. Tingkat penggunaan yang tinggi tersebut menuntut respon yang baik dan cepat. Analisis sentimen pada tugas akhir ini berguna agar bot dapat merespon sesuai dengan sentimen pengguna dan microservice memudahkan dalam peningkatan respon dari layanan.
Keterkaitan	Penggunaan microservice dalam pembangunan rancang bangun aplikasi.
Nama Peneliti	Dedy Puji Jayanto [9]
Judul Penelitian	Rancang Bangun Back-end Aplikasi “SIAP: Sistem Informasi Aspirasi dan Pengaduan Masyarakat” berbasis Web/Sms dengan menggunakan Metode Microservice Spring Boot
Deskripsi	Penelitian ini dilakukan untuk membuat aplikasi yang digunakan untuk mengimplementasikan E-government berupa Sistem Informasi Aspirasi dan Pengaduan Masyarakat. Penerapan microservice pada pembangunan rancang bangun aplikasi ini guna membagi banyak bagian aplikasi pada fungsionalitasnya.
Keterkaitan	Penggunaan microservice dalam pembangunan rancang bangun aplikasi.
Nama Peneliti	Hatma Suryotrisongko [18]

Judul Penelitian	Arsitektur Microservice untuk Resiliensi Sistem Informasi
Deskripsi	Penelitian ini dilakukan untuk mengembangkan aplikasi yang dapat meningkatkan kualitas software pada aspek resiliensinya dengan menerapkan arsitektur microservice.
Keterkaitan	Penerapan arsitektur microservice pada pembangunan rancang bangun aplikasi.

2.2 Microservice

Microservice adalah sebuah arsitektur perangkat lunak dengan paradigma penyusunan sebuah fungsionalitas menjadi *service* yang independen [10]. Arsitektur microservice berkebalikan dengan arsitektur monolitik yakni aplikasi yang terpusat dan teknologi yang seragam [11]. Arsitektur microservice membagi service sehingga setiap *service* dapat dibangun oleh tim yang berbeda dan dilakukan pengujian secara independen seperti pada gambar 2.1.



Gambar 2.1 Diagram Arsitektur *microservice*

Pembagian fungsionalitas microservice menjadi sejumlah *service* memberikan akses secara cepat tanpa terlalu membebani server [10]. *Loosely coupled* dapat membuat *service* yang ada digunakan oleh pihak eksternal dengan melihat service yang ada dan sesuai. Dengan demikian, apabila pihak pemerintah atau UMKM memiliki sistem tersendiri dan ingin mengintegrasikan sistem tersebut dengan salah satu service yang ada pada sistem

ini, maka tidak akan mengganggu service lain pada aplikasi E-inkubator.

Microservice sendiri memiliki ciri khusus yakni memiliki banyak aplikasi kecil yang saling berkomunikasi satu dengan yang lain. Aplikasi tersebut dapat dikembangkan dengan teknologi dan bahasa pemrograman yang berbeda-beda, sehingga dapat mempermudah pengembang dalam melakukan *scaling* terhadap aplikasi apabila terjadi perubahan. Dalam implementasinya, Netflix merupakan satu contoh perusahaan besar yang sukses dalam mengimplementasikan microservice.

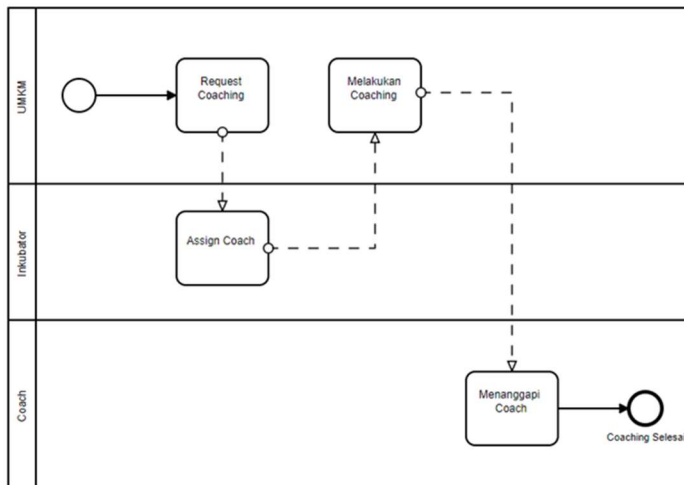
Dalam melakukan uji coba microservice, terdapat dua metode yang digunakan yaitu *Functional Testing* dan *Non-Functional Testing*. *Functional Test* dapat dilakukan melalui *whitebox testing* atau *blackbox testing*. *Non-Functional Testing* juga dapat dilakukan dengan berbagai cara sesuai dengan kebutuhan seperti keamanan, *performance*, dan lain sebagainya. Dalam melakukan *performance testing*, terdapat berbagai cara yaitu *load testing* dan *stress testing*.

2.3 Laravel/Lumen

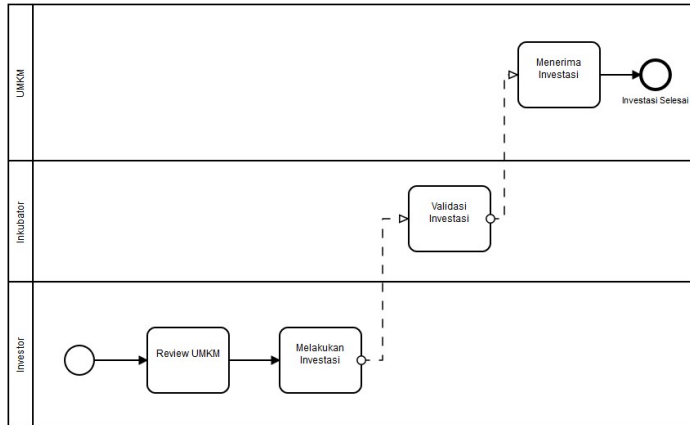
Laravel adalah suatu framework bahasa pemrograman PHP. Framework Laravel dikhususkan untuk pengembangan aplikasi web. Laravel dibangun berdasarkan paradigma MVC (Model, View, Controller) [12]. Laravel memiliki beberapa fitur utama seperti Modularity, Routing, Query-builder, dan Authentication. Modularity mengindikasikan bahwa Laravel memiliki modul-modul yang dapat digunakan untuk beberapa kebutuhan. Routing pada Laravel dapat mempermudah pengguna dalam menerapkan MVC pada pengembangan aplikasi web. Query-builder mempermudah dalam menyusun query yang nantinya akan berhubungan dengan database. Authentication mempermudah pengembang dalam membuat autentikasi pada aplikasi yang dibuat. Sedangkan Lumen sendiri merupakan *micro-framework*/turunan dari Laravel yang berfokus kepada pembuatan aplikasi kecil yang digunakan untuk membuat *microservice*.

2.4 E-Inkubator

Aplikasi E-inkubator merupakan aplikasi web yang memiliki beberapa fungsi. Fungsi-fungsi tersebut menghubungkan 4 jenis user, yaitu Inkubator, Investor, UMKM, dan Coach. Fungsi utama yang dimiliki adalah fungsi inkubasi yang dilakukan untuk meningkatkan bisnis dari UMKM dengan mengikuti Coach yang ada dalam aplikasi tersebut. Fungsi kedua yakni investasi. Pada fungsi investasi, UMKM menerima investasi oleh Investor yang memilih UMKM tersebut untuk diinvestasi agar bisnisnya dapat berkembang. Investor dalam memilih UMKM yang didanai dapat melihat riwayat keuangan UMKM dan/atau *rating* yang telah diberikan oleh Coach setelah melakukan inkubasi. Pada gambar 2.2 menggambarkan mengenai proses bisnis dari fungsi Inkubasi. Sedangkan pada gambar 2.3 menggambarkan proses bisnis dari fungsi Investasi.



Gambar 2.2 Diagram Proses Bisnis Inkubasi



Gambar 2.3 Diagram Proses Bisnis Investasi

2.4.1 UMKM

Usaha Mikro, Kecil, dan Menengah merupakan jenis usaha dimiliki perorangan atau sekelompok orang/badan usaha yang memiliki dasar hukum berupa undang-undang. Tujuan dibentuknya UMKM adalah meningkatkan perekonomian dalam negeri dengan meningkatkan usaha dan memanfaatkan tenaga kerja yang ada pada daerah sekitar [13]. Dalam aplikasi E-inkubator, UMKM berperan sebagai user yang membutuhkan bimbingan dari Coach melalui layanan Inkubasi dan mendapatkan investasi dari Investos melalui layanan Investasi.

2.4.2 Coach

Coach merupakan orang atau sekelompok orang yang ahli/expert pada bidang tertentu dalam kehidupan berbisnis, seperti membuat makanan, membuat kerajinan, desain, dan lain sebagainya. Dalam aplikasi E-inkubator, Coach berperan untuk melatih UMKM dengan melakukan pelatihan baik itu berupa tugas yang harus dikerjakan oleh UMKM atau pelatihan langsung ke UMKM tersebut. Setelah melatih UMKM tersebut, Coach dapat menilai UMKM tersebut dengan memberi *rating*. *Rating* tersebut akan menjadi salah satu faktor bahwa UMKM tersebut layak untuk diinvestasi.

2.4.3 Investor

Investor merupakan perorangan atau badan usaha yang melakukan investasi terhadap bidang usaha tertentu. Pada aplikasi E-inkubator, Investor bisa merupakan Pemerintah Pusat, Pemerintah Daerah, atau Perusahaan. Investor dalam aplikasi ini memiliki fungsi yang sama seperti investor pada umumnya yakni melakukan investasi terhadap UMKM. Investor dapat menawarkan kesepakatan terhadap UMKM apabila bisnisnya diinvestasi dengan melakukan kesepakatan tersebut sebelum menginvestasinya.

2.4.4 Inkubator

Inkubator merupakan suatu jenis bisnis yang memiliki tujuan sebagai wadah untuk suatu bisnis dapat berkembang. Inkubator dalam aplikasi E-inkubator memiliki fungsi untuk mempertemukan UMKM yang meminta inkubasi dengan Coach yang cocok dengan bisnis UMKM tersebut.

2.5 *Waterfall* SDLC

Pengembangan perangkat lunak memerlukan suatu metodologi dalam proses pengembangannya. Hal ini disebut sebagai *Software Development Life Cycle* (SDLC). Metodologi ini digunakan sebagai petunjuk untuk mengatur bagaimana seharusnya dan alur dari suatu perkembangan perangkat lunak. Salah satu model dalam SDLC adalah Waterfall. Metode Waterfall merupakan SDLC yang memiliki alur/urutan dimulai dari analisa kebutuhan, desain, tahapan implementasi desain, dan pengujian yang dikerjakan secara bertahap. Pada pengembangan aplikasi E-inkubator ini akan digunakan metode waterfall.

2.6 Unified Modeling Language (UML)

UML adalah bahasa yang menjadi standar dalam pemrograman untuk visualisasi, perancangan, dan pendokumentasian sistem perangkat lunak [14]. Pengembangan perangkat lunak dapat dilihat dari beberapa sudut pandang untuk mendapatkan gambaran perangkat lunak yang lebih baik dan menyeluruh.

Oleh karenanya, UML menyediakan beberapa macam diagram yang dapat digunakan untuk menggambarkan sudut pandang yang berbeda dari perangkat lunak.

2.7 Blackbox Testing

Blackbox testing merupakan metode pengujian untuk mengetahui fungsionalitas dari tiap fungsi pada aplikasi tanpa mengetahui proses yang ada didalamnya [15]. Pengujian ini dilakukan dengan membuat test case yang memiliki atribut masukan, keluaran, dan hasil yang diharapkan. Pengujian ini bisa dilakukan untuk aplikasi yang tidak memiliki algoritma pemrograman atau granularitasnya rendah [16], sehingga mempercepat waktu pengujian [17].

2.8 Load Testing

Load Testing merupakan metode yang digunakan untuk menguji coba seberapa tinggi tingkat *request* yang dapat diterima oleh suatu web dalam satu waktu. Dalam Load Testing, dapat diketahui *Request Per Second* dari suatu web yang apabila suatu web memiliki tingkat RPS yang tinggi maka semakin baik.

2.9 Stress Testing

Stress Testing merupakan metode yang menyerupai dengan Load Testing. Namun pada stress testing, request yang diminta lebih tinggi karena pada stress testing bertujuan untuk melakukan uji coba seberapa besar tingkat bertahan dari suatu web terhadap request yang tinggi.

Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Pada bab ini, penulis menjelaskan mengenai metodologi yang akan digunakan sebagai pedoman dalam penyelesaian tugas akhir ini.

3.1 Tahapan Pelaksanaan Tugas Akhir

3.1.1 Studi Literatur

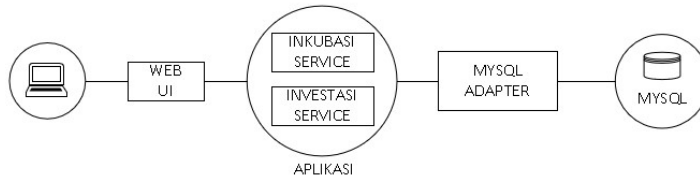
Pada tahapan ini dilakukan pengumpulan dan pengkajian pustaka dari kepustakaan yang ada tentang konsep/teori dasar dan metode pengerjaan yang digunakan untuk menyelesaikan permasalahan yang diangkat pada tugas akhir ini. Konsep dan metode yang akan digunakan adalah microservice, Aplikasi E-inkubator, Laravel/Lumen, dan SDLC Waterfall.

3.1.2 Penggalian Kebutuhan

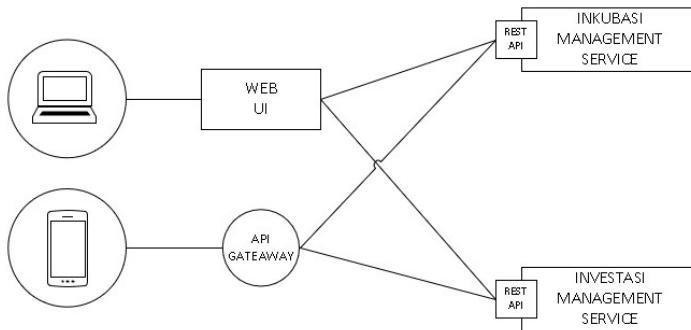
Pada tahap ini akan dilakukan penggalian kebutuhan mengenai aplikasi yang akan dikembangkan. Penggalian kebutuhan ini menghasilkan luaran berupa functional requirement dan non-functional requirement pada aplikasi E-inkubator. Functional Requirement nantinya akan dianalisis menjadi kebutuhan dari microservice.

3.1.3 Desain Sistem

Pada tahap ini dilakukan perancangan dan desain aplikasi E-inkubator. Perancangan dan desain menggunakan Unified Model Language (UML) dan desain transformasi dari arsitektur monolitik pada gambar 3.1 ke arsitektur microservice pada gambar 3.2. UML yang akan digunakan merupakan Use Case Diagram, Class Diagram, dan Component Diagram.



Gambar 3.1 Arsitektur Monolitik E-Inkubator



Gambar 3.2 Arsitektur Microservice E-Inkubator

Pada gambar dapat dilihat bahwa aplikasi monolitik pada gambar 3.2 memiliki 1 basis data utama yang dipakai oleh seluruh service dalam satu aplikasi. Namun, ketika sudah migrasi ke arsitektur microservice seperti pada gambar 3.2, database tersebut dapat dipecah menjadi kebutuhan yang berbeda-beda baik itu dari segi bahasa pemrograman yang digunakan, peran dari pengguna (*role*), maupun *service* utama yang ada.

3.1.4 Implementasi Desain Sistem

Implementasi desain sistem merupakan tahapan migrasi dari aplikasi yang awalnya monolitik, menjadi aplikasi yang berbasis microservice. Tahapan ini merupakan tahap pengkodean ulang dari Bahasa pemrograman yang digunakan dan kode *database*. Kode aplikasi dibuat dengan menggunakan framework PHP yaitu Laravel, khususnya Lumen.

3.1.5 Testing

Pengujian perangkat lunak yang akan diimplementasi adalah pengujian menggunakan *blackbox testing* untuk mengetahui jalannya fungsionalitas tiap fungsi dalam aplikasi. Blackbox testing juga dilakukan untuk memastikan aplikasi berjalan memenuhi kebutuhan bisnis dan berjalan dengan baik atau tidak ada bug dan error.

3.1.6 Deployment

Pada tahap ini, Aplikasi E-inkubator yang telah berhasil melewati serangkaian testing, dipindahkan ke server local untuk dilakukan uji coba terhadap beberapa *environment*. Setelah dipindahkan dilakukan serangkaian pengujian untuk mengklarifikasi bahwa aplikasi yang sudah dipindahkan dapat digunakan dan memiliki fungsionalitas seperti sebelum dipindahkan.

3.1.7 Dokumentasi

Pada tahap akhir ini, akan dilakukan dokumentasi berupa pembuatan laporan dalam bentuk buku tugas akhir sesuai dengan pedoman pembuatan buku tugas akhir. Buku ini berisi mengenai proses pengerjaan tugas akhir sampai selesai secara terperinci.

3.2 Jadwal Kegiatan

Pada table 3.1 merupakan waktu kegiatan dari pengerjaan tugas akhir ini.

BAB IV

ANALISIS KEBUTUHAN DAN PERANCANGAN

Pada bab ini diuraikan analisis yang diperlukan untuk kebutuhan dan perancangan sistem. Bab ini meliputi gambaran umum sistem, analisis kebutuhan sistem, dan desain sistem.

4.1 Gambaran Umum

Aplikasi E-Inkubator memiliki fungsi utama sebagai platform penyedia layanan pelatihan secara daring untuk ukm dan penyedia dana bagi ukm oleh investasi dari investor. Aplikasi ini memiliki 4 jenis pengguna/user yaitu inkubator, investor, pelatih, dan ukm. Setiap jenis user memiliki otorisasi yang berbeda sehingga hanya pengguna tertentu yang boleh melakukan fungsi tertentu.

4.2 Analisis Kebutuhan Sistem

Dalam melakukan tahap pengkodean aplikasi, sebelumnya harus dilakukan analisis terhadap kebutuhan sistem yang akan dibangun. Analisis ini dilakukan dengan melihat setiap fungsi dari proses bisnis. Analisis kebutuhan sistem meliputi analisis kebutuhan fungsional, analisis kebutuhan microservice, dan analisis kebutuhan non fungsional.

4.2.1 Analisis Kebutuhan Fungsional

Dalam aplikasi E-Inkubator, dibutuhkan fungsi seperti pada tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

No.	Kebutuhan Fungsional	Deskripsi	Actor
1.	Login	User masuk menggunakan data kredensial berupa email dan sandi	Semua User
2.	Melihat Profil	User melihat profil diri	Semua User
3	Logout	User melakukan logot dari sistem	Semua User

No.	Kebutuhan Fungsional	Deskripsi	Actor
3.	Memperbaharui Profil	User memperbaharui profil diri	Semua User
4.	Coaching	User melakukan pelatihan dengan metode chatting	Coach, UMKM
5.	Melihat Riwayat Coaching	User dapat melihat riwayat pelatihan pada histori chatnya	Coach, UMKM
4.	Menerima Investasi	User menerima bukti investasi yang diberikan	UMKM
7.	Melakukan Investasi	User mengisi form investasi	Investor
9.	Melihat List UMKM	User melihat daftar UMKM yang terdaftar pada sistem	Investor
10.	Melakukan Verifikasi Investasi	User melakukan verifikasi terhadap investasi yang sudah diisi	Inkubator
11.	Melakukan Manejemen User	User melakukan manajemen terhadap user yang ada	Inkubator
12.	Melakukan Manajemen Investasi	User melakukan manajemen terhadap investasi yang sudah ada	Inkubator
13.	Melakukan Manajemen Coaching	User melakukan manajemen terhadap coaching yang ada ataupun belum ada	Inkubator

4.2.2 Analisis Kebutuhan Microservice

Dari kebutuhan fungsional yang sudah dianalisis, Kebutuhan Fungsional kemudian diturunkan menjadi kebutuhan microservice sesuai dengan aktor yang melakukan. Pada implementasi, pemecahan kebutuhan ini akan menjadi faktor yang menentukan berapa banyak *service* yang akan dibuat.

a. UMKM

Dari kebutuhan fungsional peran UMKM kemudian diturunkan menjadi kebutuhan microservice UMKM seperti pada gambar 4.2.

Tabel 4.2 Kebutuhan Microservice UMKM

Kebutuhan Fungsional	Kebutuhan Microservice
Login	Login User UMKM
Melihat Profil	Mengambil data profil
Memperbaharui Profil	Melakukan update data profil
Logout	Logout User
Coaching	Mengambil data coaching
	Mengirim data coaching
Melihat Riwayat Coaching	Mengambil data riwayat coaching
Menerima Investasi	Mengambil data investasi

b. Coach

Dari kebutuhan fungsional peran Coach kemudian diturunkan menjadi kebutuhan microservice Coach seperti pada gambar 4.3.

Tabel 4.3 Kebutuhan Microservice Coach

Kebutuhan Fungsional	Kebutuhan Microservice
Login	Login User Coach
Melihat Profil	Mengambil data profil
Memperbaharui Profil	Melakukan update data profil
Logout	Logout User
Coaching	Mengambil data coaching
	Mengirim data coaching

c. Investor

Dari kebutuhan fungsional peran Investor kemudian diturunkan menjadi kebutuhan microservice Investor seperti pada gambar 4.4.

Tabel 4.4 Kebutuhan Microservice Investor

Kebutuhan Fungsional	Kebutuhan Microservice
Login	Login User Investor
Melihat Profil	Mengambil data profil
Memperbaharui Profil	Melakukan update data profil
Logout	Logout User
Melakukan Investasi	Mengambil data histori investasi
	Membuat data investasi baru
Melihat List UMKM	Mengambil data user UMKM

d. Inkubator

Dari kebutuhan fungsional peran Inkubator kemudian diturunkan menjadi kebutuhan microservice UMKM seperti pada gambar 4.5.

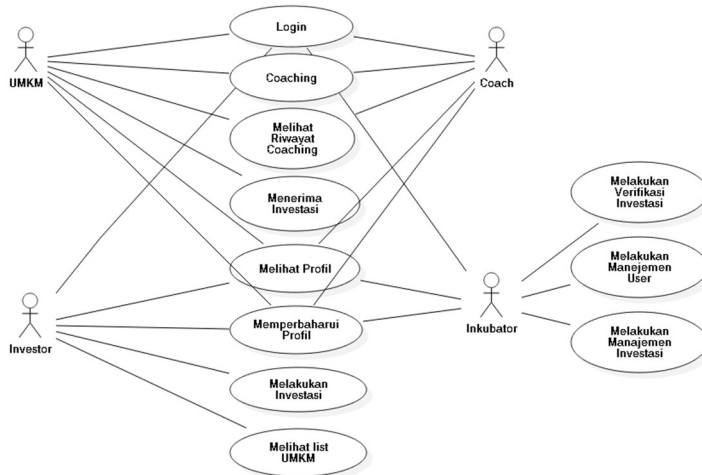
Tabel 4.5 Kebutuhan Microservice Inkubator

Kebutuhan Fungsional	Kebutuhan Microservice
Login	Login User Admin
Melihat Profil	Mengambil data profil
Memperbaharui Profil	Melakukan update data profil
Logout	Logout User
Melakukan Verifikasi Investasi	Mengambil data investasi
	Melakukan update status investasi
Melakukan Manajemen User	Mengambil data user
	Melakukan update data user
	Melakukan register data user
	Menghapus data user
Melakukan Manajemen Investasi	Mengambil data investasi
	Melakukan update data investasi
	Menghapus data investasi
Melakukan Manajemen Coaching	Mengambil data coaching
	Membuat data coaching
	Melakukan update data coaching

4.3 Desain Sistem

4.3.1 Use Case Diagram

Dalam mendesain kebutuhan sistem yang sudah dianalisis, didesain use case diagram yang menggambarkan fungsi secara umum pada sistem



Gambar 4.1 Use Case Diagram

Dalam memahami use case diagram lebih dalam seperti pada gambar 4.1, dibuat Use Case Description untuk menjelaskan detail dari setiap use case.

4.3.2 Use Case Description

1. Login

Fungsi login memiliki use case description seperti pada tabel 4.6.

Tabel 4.6 Use Case Description Login

Nama use case	Login	
Deskripsi	User dapat melakukan login untuk masuk ke akun yang telah dibuat sebelumnya	
Tujuan	Agar user dapat masuk dan dapat menggunakan fitur-fitur pada aplikasi	
Aktor	Pelaku UKM, Industri dan Pemerintah, Coach	
Trigger	User ingin masuk kedalam aplikasi dan menggunakan fitur-fitur	
Pre-condition	User telah mendaftar sebuah akun	
Basic Flow	Aktor	System
	1. Klik tombol Login	
		2. Menampilkan Login Form
	3. Memasukkan Username, dan password	
	4. Klik tombol Login	
		5. Memverifikasi data
		6. Menampilkan tampilan beranda
Alternate Flow	User memasukkan email yang tidak cocok dengan password atau sebaliknya.	
Post Condition on succes	User berhasil masuk sesuai dengan peran	
Post Condition on failure	User tidak berhasil masuk	

2. Melihat Profil

Fungsi melihat profil memiliki use case description seperti pada tabel 4.7.

Tabel 4.7 Use Case Description Melihat Profil

Nama use case	Melihat Profil	
Deskripsi	User melihat profil diri	
Tujuan	User dapat mengetahui profil diri	
Aktor	Semua user	
Trigger	-	
Pre-condition	Sudah login	
Basic Flow	Aktor	Sistem
	Membuka menu Profil	
		Menampilkan menu profil
Alternate Flow	-	
Post Condition on succes	Data tampil	
Post Condition on failure	-	
Information Need	Data profil user	

3. Memperbaharui Profil

Fungsi memperbaharui profil memiliki use case description seperti pada gambar 4.8.

Tabel 4.8 Use Case Description Memperbaharui Profil

Nama use case	Memperabahrui Profil	
Deskripsi	User memperbaharui profil	
Tujuan	User dapat memperbaharui profil diri	
Aktor	Semua user	
Trigger	-	
Pre-condition	Sudah login	
Basic Flow	Aktor	Sistem
	Membuka menu Profil	
		Menampilkan menu profil

	Menekan tombol Edit	
		Menampilkan form edit
	Melakukan edit	
	Menekan simpan data	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Data tersimpan	
Post Condition on failure	Data tidak tersimpan	
Information Need	Data profil user	

4. Berdiskusi dengan Coach

Fungsi berdiskusi dengan caoch memiliki use case description seperti pada gambar 4.9.

Tabel 4.9 Use Case Description Berdiskusi dengan Coach

Nama use case	Berdiskusi dengan Coach	
Deskripsi	Roomchat antara pelaku UKM dengan Coach dalam melakukan konsultasi, pendampingan, dll.	
Tujuan	Agar pelaku UKM dapat berkomunikasi secara intens dengan coach	
Aktor	Pelaku UKM	
Trigger	Pelaku UKM ingin berkonsultasi dengan coach	
Pre-condition	-	
Basic Flow	Pelaku UKM	Sistem
	Membuka menu E-coaching	
		Menampilkan roomchat dengan coach
	Mengisi field percakapa	

	Menekan tombol kirim	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Pesan terkirim	
Post Condition on failure	Pesangagal terkirim	
Information Need	Percakapan masalah dan solusi (Untuk data pencarian)	

5. Melihat riwayat coaching

Fungsi melihat riwayat coaching memiliki use case description seperti pada gambar 4.10.

Tabel 4.10 Use Case Description Melihat Riwayat Coaching

Nama use case	Melihat riwayat coaching	
Deskripsi	Roomchat antara pelaku UKM dengan Coach dalam melakukan konsultasi, pendampingan, dll.	
Tujuan	Agar pelaku UKM dapat melihat riwayat chat	
Aktor	Pelaku UKM	
Trigger	-	
Pre-condition	-	
Basic Flow	Pelaku UKM	Sistem
	Membuka menu history coaching	
		Menampilkan history dengan coach
Alternate Flow	-	
Post Condition on succes	Data tampil	

Post Condition on failure	
Information Need	Percakapan masalah dan solusi

6. Menerima Investasi

Fungsi menerima investasi memiliki use case description seperti pada gambar 4.11.

Tabel 4.11 Menerima Investasi

Nama use case	Menerima Investasi	
Deskripsi	User menerima invesasi dari investor	
Tujuan	User dapat melihat histori dan status investasi	
Aktor	Pelaku UKM	
Trigger		
Pre-condition	-	
Basic Flow	Pelaku UKM	Sistem
	Membuka menu investasi	
		Menampilkan history investasi
	Melihat detail investasi	
Alternate Flow	-	
Post Condition on success	Data tampil	
Post Condition on failure		
Information Need	Data Investasi	

7. Coach menanggapi pesan UKM

Fungsi menanggapi pesan ukm memiliki use case description seperti pada gambar 4.12.

Tabel 4.12 Menanggapi Pesan UKM

Nama use case	Menanggapi pesan UKM	
Deskripsi	Roomchat antara pelaku UKM dengan Coach dalam melakukan konsultasi, pendampingan, dll.	
Tujuan	Agar pelaku UKM dapat berkomunikasi secara intens dengan coach	
Aktor	Coach	
Trigger	Pelaku UKM ingin berkonsultasi dengan coach	
Pre-condition	-	
Basic Flow	Coach	Sistem
	Membuka menu roomchat	
		Menampilkan roomchat dengan UKM
	Mengakses histori percakapan	
	Mengisi field percakapan	
	Menekan tombol kirim	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Pesan terkirim	
Post Condition on failure	Pesangagal terkirim	
Information Need	Percakapan masalah dan solusi	

8. Mencari data masalah dan solusi

Fungsi mencari data maslaah dan solusi memiliki use case description seperti pada gambar 4.13.

Tabel 4.13 Use Case Description Mencari Data Masalah dan Solusi

Nama use case	Melakukan Pencarian Topik Masalah dan Solusi	
Deskripsi	Fitur pada roomchat yang dapat diakses oleh Coach ketika melakukan percakapan. Coach dapat mengakses data histori percakapan yang berisi masalah dan solusi.	
Tujuan	Agar coach dapat dimudahkan ketika menghadapi masalah yang sama dengan masalah yang sudah terselesaikan sebelumnya	
Aktor	Coach	
Trigger	Untuk membantu coach mencari solusi atas permasalahan	
Pre-condition	Coach sebelumnya melakukan input data terhadap solusi dan permasalahan dalam percakapannya dengan UKM untuk database sistem	
Basic Flow	Aktor	Sistem
	Membuka menu List Solusi	
		Menampilkan data masalah dan solusi
	Klik menu pencarian	
		Menampilkan menu Pencarian
	Memilih kategori masalah dan solusi, memasukkan kata kunci	
		Klik tombol Lihat
	Menampilkan data histori percakapan	
Alternate Flow	Masalah dan Solusi belum pernah terdeteksi	
Post Condition on succes	Menemukan permasalahan dan solusi yang sesuai	
Post Condition on failure	Data tidak dapat ditemukan	

Information Need	-
------------------	---

9. Melakukan Investasi

Fungsi melakukan investasi memiliki use case description seperti pada gambar 4.14.

Tabel 4.14 Use Case Description Melakukan Investasi

Nama use case	Melakukan Investasi	
Deskripsi	Menu untuk investor dapat melakukan investasi ke UKM yang diinginkan	
Tujuan	Memberikan akses investor untuk transaksi investasi	
Aktor	Industri dan Pemerintah	
Trigger	Tertarik melakukan investasi kepada UKM yang diinginkan	
Pre-condition	Memiliki akun industri atau pemerintah	
Basic Flow	Aktor	Sistem
	Klik menu investasi	Menampilkan halaman investasi
	Memilih pilihan investasi, memilih jumlah investasi	
	Klik tombol Transaksi Investasi	
		Menampilkan informasi investasi
	Mengupload bukti transfer	
		Verifikasi data transfer
Alternate Flow	-	
Post Condition on succes	Data Tersimpan	
Post Condition on failure	-	

Information Need	Data Investasi, bukti Investasi
------------------	---------------------------------

10. Melihat histori investasi

Fungsi melihat histori investasi memiliki use case description seperti pada gambar 4.15.

Tabel 4.15 Use Case Description Melihat Histori Investasi

Nama use case	Melihat histori investasi	
Deskripsi	Menu untuk investor dapat melihat histori investasi yang telah dilakukan	
Tujuan	untuk dapat melihat histori investasi yang telah dilakukan	
Aktor	Industri dan Pemerintah	
Trigger	-	
Pre-condition	Memiliki akun industri atau pemerintah	
Basic Flow	Aktor	Sistem
	Klik menu histori investas	
		Menampilkan histori investasi
	Melihat histori investasi	
Alternate Flow	-	
Post Condition on succes	-	
Post Condition on failure	-	
Information Need	Pilihan investasi, histori Investasi: Nama UKM, Jumlah investasi, tanggal investasi, status investasi (Aktif/berakhir)	

11. Melihat list UKM

Fungsi melihat list ukm memiliki use case description seperti pada gambar 4.16.

Tabel 4.16 Use Case Description Melihat List UKM

Nama use case	Melihat list UKM	
Deskripsi	Menu untuk melihat daftar ukm	
Tujuan	untuk dapat melihat daftar ukm yang bisa di investasi	
Aktor	Industri dan Pemerintah	
Trigger	-	
Pre-condition	Memiliki akun industri atau pemerintah	
Basic Flow	Aktor	Sistem
	Klik menu list UKM	
		Menampilkan list UKM
	Melihat detail UKM	
Alternate Flow	-	
Post Condition on succes	Data tampil	
Post Condition on failure	-	
Information Need	Data UKM	

12. Melakukan verifikasi investasi

Fungsi melakukan verifikasi investasi memiliki use case description seperti pada gambar 4.17.

Tabel 4.17 Use Case Description Melakukan Verifikasi Investasi

Nama use case	Melakukan verifikasi investasi
Deskripsi	Menu untuk melakukan verifikasi investasi

Tujuan	untuk dapat memverifikasi investasi yang sudah dilakukan oleh investor	
Aktor	Inkubator	
Trigger	-	
Pre-condition	Login	
Basic Flow	Aktor	Sistem
	Klik menu investasi	
		Menampilkan list investasi
	Melakuakn edit status investasi	
	Menyimpan hasil perubahan	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Data tersimpan	
Post Condition on failure	-	
Information Need	Data Investasi	

13. Melakukan manajemen user

Fungsi melakukan manajemen user memiliki use case description seperti pada gambar 4.18.

Tabel 4.18 Use Case Description Melakukan Manajemen User

Nama use case	Melakukan manajemen user
Deskripsi	Melakukan CRUD terhadap data user
Tujuan	Untuk dapat melakukan CRUD terhadap data user
Aktor	Inkubator

Trigger	-	
Pre-condition	Login	
Basic Flow	Aktor	Sistem
	Klik menu User Management	
		Menampilkan data user
	Melakukan CRUD data user	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Data tersimpan	
Post Condition on failure	-	
Information Need	Data User	

14. Melakukan manajemen investasi

Fungsi manajemen investasi memiliki use case description seperti pada gambar 4.19.

Tabel 4.19 Use Case Description Melakukan Manajemen Investasi

Nama use case	Melakukan manajemen investasi	
Deskripsi	Melakukan RUD terhadap data investasi	
Tujuan	Untuk dapat melakukan RUD terhadap data investasi	
Aktor	Inkubator	
Trigger	-	
Pre-condition	Login	
Basic Flow	Aktor	Sistem
	Klik menu Investasi Management	

		Menampilkan data investasi
	Melakukan RUD data investasi	
		Menyimpan data
Alternate Flow	-	
Post Condition on succes	Data tersimpan	
Post Condition on failure	-	
Information Need	Data Investasi	

15. Melakukan manajemen Coaching

Fungsi manajemen coaching memiliki use case description seperti pada gambar 4.20.

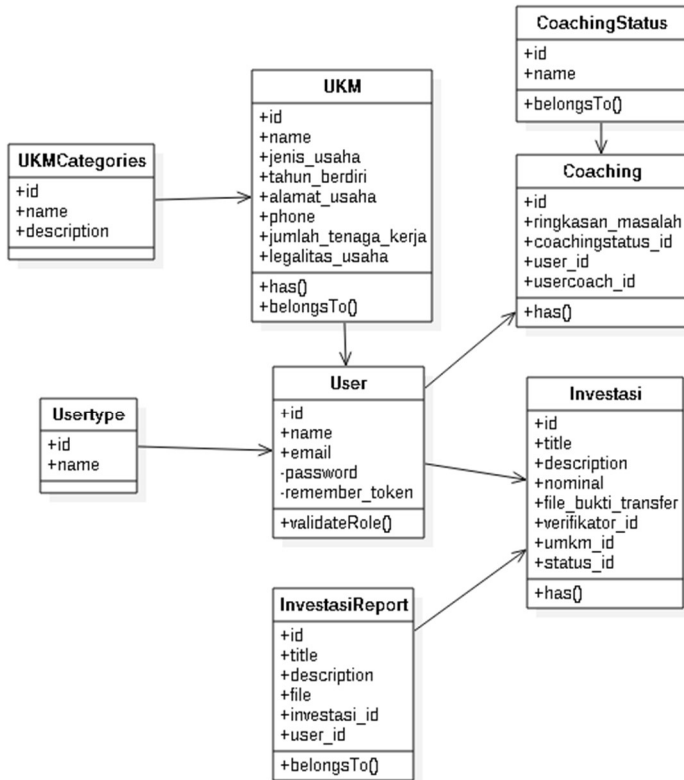
Tabel 4.20 Use Case Description Manajemen Coaching

Nama use case	Melakukan manajemen coaching	
Deskripsi	Melakukan CRUD terhadap data coaching	
Tujuan	Untuk dapat melakukan CRUD terhadap data coaching	
Aktor	Inkubator	
Trigger	-	
Pre-condition	Login	
Basic Flow	Aktor	Sistem
	Klik menu Coaching Management	
		Menampilkan data coaching
	Melakukan CRUD data coaching	
		Menyimpan data

Alternate Flow	-
Post Condition on succes	Data tersimpan
Post Condition on failure	-
Information Need	Data Coaching

4.3.3 Class Diagram

Dari use case yang telah dibuat pada pembahasan sebelumnya, dibuat class diagram seperti pada gambar 4.2 yang nantinya akan menjadi *class* pada kode aplikasi.

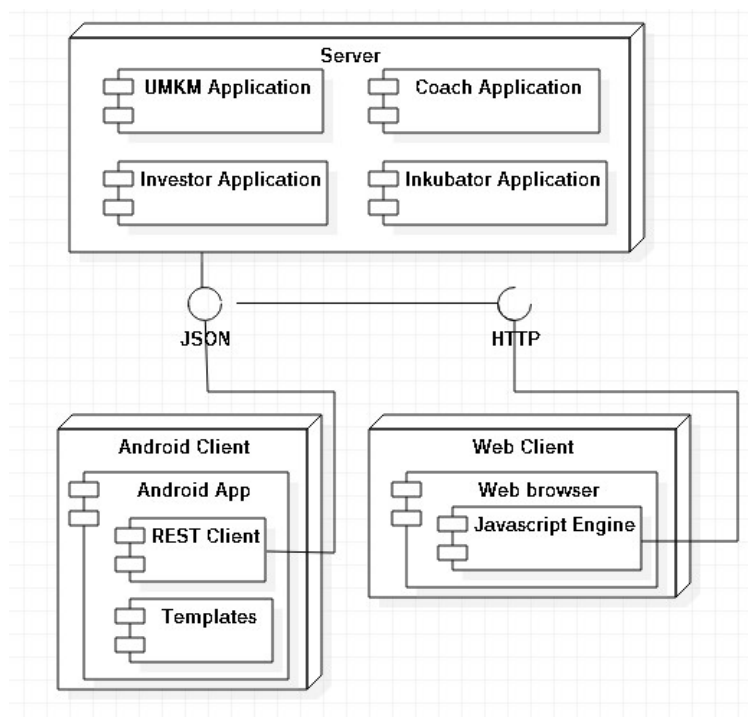


Gambar 4.2 Class Diagram

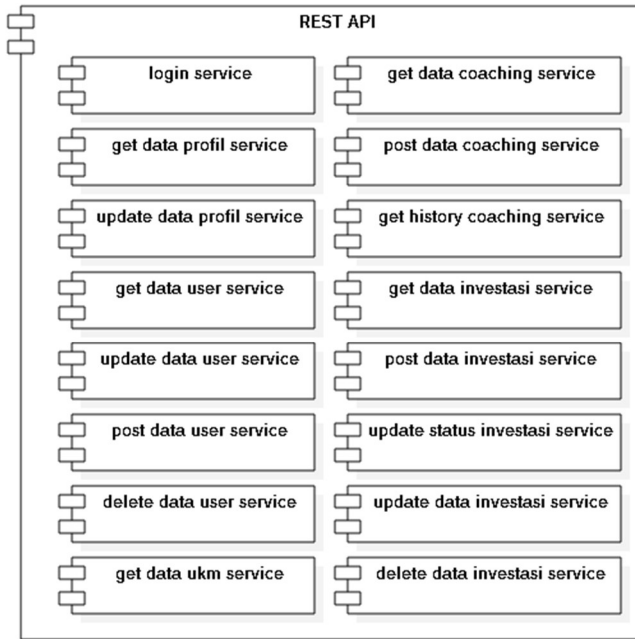
4.3.4 Component Diagram

Berikut Component Diagram yang akan dibuat dalam penggunaan microservice. Component diagram pada gambar 4.3 menggambarkan arsitektur dari microservice yang dibuat

memiliki beberapa service kecil yang terpisah satu sama lain dan memiliki database tersendiri. Service yang terpisah tersebut dipecah berdasarkan peran pengguna yaitu, UMKM, Coach, Investor, dan Inkubator.



Gambar 4.3 Component Diagram

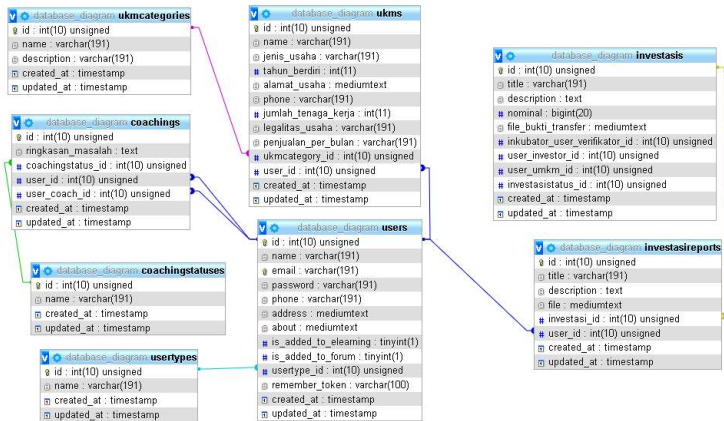


Gambar 4.4 Detail Component Daigram

Pada gambar 4.4 menggambarkan tentang detail dari component diagram yang berisi mengenai service apa saja yang ada pada aplikasi E-Inkubator.

4.3.5 Desain Database

Desain Database yang akan dibuat pada aplikasi ini akan digunakan untuk menyimpan data yang ada pada aplikasi. Pada gambar 4.5 menggambarkan desain database yang akan dibuat.



Gambar 4.5 Desain Database

Dari desain database yang dibuat, dilakukan pemecahan terhadap database termasuk *dependency*-nya sesuai dengan kebutuhan microservice yang sudah didefinisikan sebelumnya sehingga tabel yang ada pada database dari setiap service digambarkan seperti pada tabel .

Table 4.1 Pemecahan Tabel Basis Data

No	Service	Tabel
1	UMKM	Ukm, Ukmcategories, Users
2	Coach	Coachings, Caochingstatuses, Users
3	Investor	Investasis, Investasirpeort, Users
4	Inkubator	Coachings, Ukm, Investasi, Users, Usertype

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

5.1 Lingkungan Implementasi

Aplikasi E-Inkubator dikembangkan menggunakan perangkat keras komputer. Spesifikasi perangkat keras yang digunakan dapat dilihat pada tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Perangkat Keras	Spesifikasi	
PC/Laptop	Prosesor	Intel Core i5 6200
	RAM	4096 MB

Dalam mengembangkan aplikasi ini, juga menggunakan beberapa perangkat lunak. Spesifikasi Perangkat lunak tersebut seperti pada tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Versi
Sistem Operasi	Windows 10 Education
Web Server	Apache 2.4.29
Basis Data	MySQL 5.0.12
Bahasa Pemrograman	PHP 7.2.0
Text Editor	PhpStorm 2018

5.2 Implementasi Desain Aplikasi

Dalam implementasi desain aplikasi, dibuat menggunakan Lumen yang merupakan *framework* turunan dari Laravel yang mempermudah pengguna dalam mengimplementasikan *microservice*. Dalam implementasi tugas akhir ini, dibuat 4 aplikasi yang menjadi *back-end* dari *microservice* dengan 1 *front-end* berupa aplikasi dengan *framework* laravel.

5.2.1 Implementasi Melihat Investasi

Fungsi investasi digunakan Investor untuk melihat investasi apa saja yang sudah dilakukannya

User Management

No	Nama	Description	Nominal	File Bukti	Target UMKM	Investor	Status	Action
1	Investasi Kerjasama Gepreker	INVESTASI 2	30000000		Geprekers - UMKM User	PT Investa Utama - Investor	Investasi Selesai	Detail
2	Investasi Kerjasama Gepreker	A	12000	924161942.jpg	Geprekers - UMKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
3	Inv 1	Investasi 1	200000	593511152.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
4	Inv 2	Invensi	123123	716254896.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
5	Inv 3	Invensi	23120000	689070374.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
6	Inv 23	Invensi	12000	197855424.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
7	Inv 4	Invensi	12000	399623413.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
8	Onv 5	Invensi	12000	990120667.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
9	Onv 5	Invensi	12000	388814385.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail

Gambar 5.1 Front-end Melihat Investasi

Untuk mendapatkan tampilan tersebut, dibuat kode program seperti pada gambar 5.2.

```
public function index()
{
    $client = new Client(); //GuzzleHttp\Client
    $response = $client->get('http://localhost/lumen-inkubator/public/investasi');
    $data = $response->getBody();
    $data = json_decode(json: $data, assoc: true);

    return view('admin.investasi.investasi', compact('data'));
}
```

Gambar 5.2 Potongan Kode Front-end Melihat Investasi

Dari keseluruhan program, kode untuk front-end mengambil client dari lumen lain untuk mengambil data yang dibutuhkan menggunakan package bernama Guzzle.

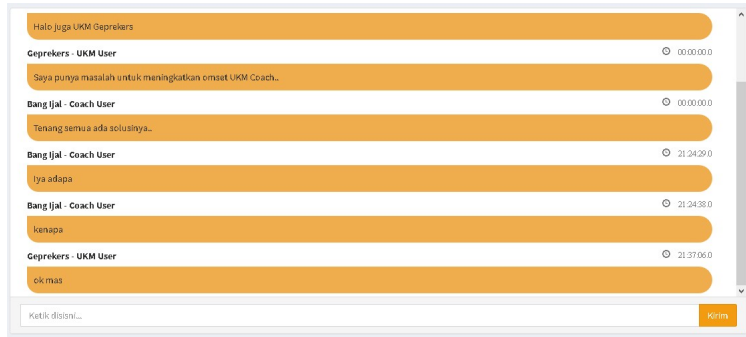
Fungsi get data investasi menggunakan microservice untuk mengambil semua data dari investasi yang sudah dilakukan oleh investor. Potongan kode dari setiap fungsi get data investasi terlihat pada gambar 5.3.

```
public function getInvestasi()
{
    $data = Investasi::all();
    return response()->json(['data'=>$data]);
}
```

Gambar 5.3 Potongan Kode Microservice Melihat Investasi

5.2.2 Implementasi inkubasi service

Fungsi inkubasi digunakan Ukm untuk melakukan chatting terhadap coach.



Gambar 5.4 Tampilan Melakukan Inkubasi

Fungsi layanan inkubasi menggunakan microservice terlihat pada gambar 5.5.

```
public function postChat(Request $request)
{
    $user_id = Auth::user()->id;
    $coachingmessage = new Coachingmessage();
    $coachingmessage->user_id = $user_id;
    $coachingmessage->coaching_id = $request->id_coaching;
    $coachingmessage->coachingmessage_id = 1;
    $coachingmessage->content = $request->content_message;
    $coachingmessage->save();

    $coaching = Coaching::where('id', $request->id_coaching)->first();
    $coaching_message = Coachingmessage::where('coaching_id', $request->id_coaching)->get();
    return response()->json($coaching_message);
}

public function getChat(Request $request)
{
    $user_id = Auth::user()->id;
    $coaching = Coaching::where('id', $request->id_coaching)->first();
    $coaching_message = Coachingmessage::where('coaching_id', $request->id_coaching)->with('user')->get();
    return response()->json($coaching_message);
}
```

Gambar 5.5 Potongan Kode Microservice Melakukan Inkubasi

5.2.3 Login

Fungsi login digunakan oleh pengguna untuk masuk kedalam aplikasi. Pada gambar 5.6 memperlihatkan tampilan login.

Gambar 5.6 Front-end Login

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.7.

```
<form class="form-horizontal" method="POST" action="{{ route('login') }}">
  {{ csrf_field() }}
  <div class="form-group"{{ $errors->has('email') ? ' has-error' : '' }}>
    <label for="email" class="col-md-4 control-label">E-mail Address</label>
    <div class="col-md-6">
      <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
      @if ($errors->has('email'))
        <span class="help-block">
          <strong>{{ $errors->first('email') }}</strong>
        </span>
      @endif
    </div>
  </div>
  <div class="form-group"{{ $errors->has('password') ? ' has-error' : '' }}>
    <label for="password" class="col-md-4 control-label">Password</label>
    <div class="col-md-6">
      <input id="password" type="password" class="form-control" name="password" required>
      @if ($errors->has('password'))
        <span class="help-block">
          <strong>{{ $errors->first('password') }}</strong>
        </span>
      @endif
    </div>
  </div>
</form>
```

Gambar 5.7 Potongan Kode Front-end Login

Fungsi login juga menggunakan microservice untuk memvalidasi setiap REST API yang nantinya akan digunakan seperti potongan kode pada gambar 5.8.

```
protected function authenticated($request, $user)
{
    if($user->hasRole(1)){
        return redirect('/admin/home');
    }else if($user->hasRole(2)){
        return redirect('/umkmuser/home');
    }else if($user->hasRole(3)){
        return redirect('/coachuser/home');
    }else if($user->hasRole(4)){
        return redirect('/investor/home');
    }
}
```

Gambar 5.8 Potongan Kode Microservice Login

5.2.4 Melihat Profil

Fungsi melihat profil digunakan pengguna untuk melihat profil diri. Pada gambar 5.10 merupakan tampilan dari profil.

Profil Coach	
Nama Coach	Bang Ijal - Coach User
Email Coach	coach@gmail.com
Phone	0322222221
Address	Coach dari Manajemen Bisnis ITS, Sukolilo, Surabaya
About	Coach handal terpercaya saja
<div> Edit Delete </div>	

Gambar 5.9 Tampilan Profil Diri

Penerapan microservice yang ada pada tampilan profil dibuat seperti pada gambar 5.10.

```

public function getProfil()
{
    $data = Auth::user();

    return response()->json(['data' => $data
    ]);
}

```

Gambar 5.10 Potongan Kode Menampilkan Profil

5.2.5 Memperbaharui profil

Fungsi memperbaharui profil berfungsi untuk pengguna yang ingin memperbaharui profil. Tampilan untuk memperbaharui profil seperti pada gambar 5.11.

Edit Profil Coach

Nama Coach:
Bang Ijal - Coach User

Email Coach:
coach@gmail.com

Phone:
0322222221

Alamat:
Coach dari Manajemen Bisnis ITS, Sukolilo, Surabaya

About:
Coach handal terpercaya saja

Submit **Cancel**

Gambar 5.11 Tampilan Front-end Memperbaharui Profil

Fungsi memperbaharui profil mengimplementasikan microservice seperti pada potongan kode pada gambar 5.12.

```

public function editProfil()
{
    $id = Auth::user()->id;
    $data = User::where('id', $id)->first();

    return response()->json(['data' => $data
    ]);
}

```

Gambar 5.12 Potongan Kode Microservice Memperbaharui Profil

5.2.6 Melihat History Coaching

Fungsi Melihat History Coaching digunakan oleh Coach dan Ukm untuk dapat melihat riwayat histori. Tampilan dari fungsi ini seperti pada gambar 5.13.



Gambar 5.13 Tampilan Front-end History Coaching

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.14.

```

public function history()
{
    $data = $this->getHistory()->getContent();
    return view( view: 'ukm.history' )->with('data', json_decode($data, assoc: true));
}

```

Gambar 5.14 Potongan Kode Front-end History Coaching

Fungsi ini menerapkan microservice seperti pada gambar 5.15.

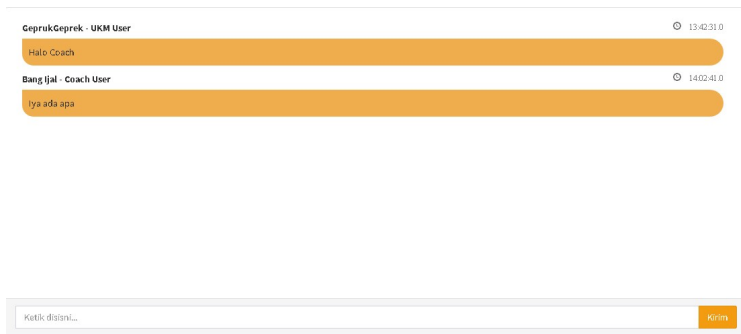
```
public function getHistory()
{
    $id = Auth::user()->id;
    $data = Coaching::where('user_id', $id)->get();

    return response()->json(['data' => $data
    ]);
}
```

Gambar 5.15 Potongan Kode Microservice History Coaching

5.2.7 Melakukan Coaching

Fungsi ini digunakan oleh Coach dan Ukm untuk melakukan chatting untuk mengatasi permasalahan Ukm. Tampilan coaching dapat terlihat pada gambar 5.16.



Gambar 5.16 Tampilan Coaching

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.17.

```
public function chat($id = null)
{
    if ($id != null) {
    } else {
        $data = $this->getChat()->getContent();
        return view('view: 'umkm.chat')->with('data', json_decode($data, assoc: true));
    }
}
```

Gambar 5.17 Potongan Kode Front-end Coaching

Fungsi Coaching menerapkan microservice seperti pada gambar 5.18.

```
public function getChat($id = null)
{
    if ($id != null) {
    } else {
        $user_id = Auth::user()->id;
        $data = Coaching::where('user_id', $user_id)->where('coachingstatus_id', 2)->first();
        return response()->json(['data' => $data]);
    }
}
```

Gambar 5.18 Potongan Kode Microservice Coaching

5.2.8 Melihat Investasi yang diterima

Fungsi melihat investasi diterima digunakan oleh ukm untuk melihat investasi apa saja yang sudah diterima. Pada gambar 5.19 merupakan tampilan melihat investasi yang diterima.

Riwayat Investasi								
No	Nama	Description	Nominal	File Bukti	Target UMKM	Investor	Status	Action
1	Investasi Kerjasama Gepreker	INVESTASI 2	30000000		Geprekers - UKM User	PT Investa Utama - Investor	Investasi Selesai	Detail
2	Investasi Kerjasama Gepreker	A	12000	924161942.jpg	Geprekers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
3	Inve	Inve	12000	98497573.jpg	Geprekers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
4	Investasi TA	A	200000	669646347.jpg	Geprekers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail

Gambar 5.19 Tampilan Front-end Investasi yang diterima

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.20.

```
public function investasilist()
{
    $data = $this->getInvestasi()->getContent();
    return view('ukm.investasilist')->with('data', json_decode($data, assoc: true));
}
```

Gambar 5.20 Potongan Kode Front-end Investasi yang diterima

Fungsi ini menerapkan microservice seperti pada gambar 5.21.

```

public function getInvestasi()
{
    $id = Auth::user()->id;
    $data = Investasi::where('user_umkm_id', $id)->get();

    return response()->json(['data' => $data
    ]);
}


```

Gambar 5.21 Potongan Kode Microservice Investasi yang diterima

5.2.9 Melihat detail Investasi yang diterima

Fungsi Detail Investasi yang diterima berfungsi untuk melihat detail dari investasi yang diterima oleh umk. Tampilan fungsi ini dapat dilihat pada gambar 5.22.

Detail Investasi

Title	Inve
Description	Inve
Nominal	12000
UMKM	Geprekers - UKM User
Verifikator	Belum terverifikasi
Status	Belum dikonfirmasi
Bukti Transfer	

Back

Gambar 5.22 Tampilan Front-end Detail Investasi

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.23.

```

public function investasidetil($id)
{
    $data = $this->getInvestasiDetail()->getContent();
    return view( view: 'umkm.investasidetil' )->with('data', json_decode($data, assoc: true));
}

```

Gambar 5.23 Potongan Kode Front-end Detail Investasi

Fungsi ini menerapkan microservice dengan potongan kode seperti pada gambar 5.24.

```

public function getInvestasiDetail($id)
{
    $data = Investasi::find($id);

    return response()->json(['data' => $data
    ]);
}

```

Gambar 5.24 Potongan Kode Microservice Detail Investasi

5.2.10 Menambahkan Investasi

Fungsi Menambahkan Investasi berfungsi untuk menambahkan investasi ke ukm oleh investor. Tampilan menambahkan investasi seperti pada gambar 5.25.

Gambar 5.25 Tampilan Front-end Menambahkan Investasi

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.26.

```

public function store(Request $request)
{
    $data = $this->postInvestasi()->getContent();

    return redirect()->route( route: 'investor.investasialist' )->with('data', json_decode($data, assoc: true));
}

```

Gambar 5.26 Potongan Kode Front-end Menambahkan Investasi

Fungsi ini menerapkan microservice dengan potongan kode pada gambar 5.27.

```

public function postInvestasi(Request $request)
{
    $data = new Investasi();
    $data->title = $request->title;
    $data->description = $request->description;
    $data->nominal = $request->nominal;
    $file = $request->file( key: 'fileToUpload' );
    $ext = $file->getClientOriginalExtension();
    $newName = rand(100000,1001238912)." ".$ext;
    $destPath = 'uploads/file';
    $file->move($destPath,$newName);

    $data->file_bukti_transfer = $newName;
    $data->inkubator_user_verifikator_id = $request->inkubator_user_verifikator_id;
    $data->user_investor_id = Auth::user()->id;
    $data->user_umkm_id = $request->user_umkm_id;
    $data->investasistatus_id = 1;
    if($data->save()){
        $res['message'] = "Success!";
        $res['value'] = "$data";
        return response($res);
    }else{
        $res['message'] = "Failed!";
        return response($res);
    }
}

```

Gambar 5.27 Potongan Kode Microservice Menambahkan Investasi

5.2.11 Melihat Investasi yang telah dilakukan

Fungsi Melihat Investasi yang dilakukan berfungsi untuk melihat investasi apa saja yang telah dilakukan oleh investor. Tampilan melihat investasi yang dilakukan dapat dilihat pada gambar 5.28.

No	Nama	Description	Nominal	File Bukti	Target UMKM	Investor	Status	Action
1	Investasi Karjasama Geprekr	INVESTASI 2	30000000		Geprekrers - UKM User	PT Investa Utama - Investor	Investasi Selesai	Detail
2	Investasi Karjasama Geprekr	A	12000	924161942.jpg	Geprekrers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
3	Inve 23	Inves	12000	197855424.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
4	Inv 4	inve	12000	299623413.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
5	Onves	INves	12000	990120667.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
6	Onves	INves	12000	388814385.jpg	Admin 1	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
7	Inve	inve	12000	98497573.jpg	Geprekrers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail
8	Investasi TA	A	200000	859645347.jpg	Geprekrers - UKM User	PT Investa Utama - Investor	Belum dikonfirmasi	Detail

Gambar 5.28 Tampilan Front-end Melihat Investasi

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.29.

```
public function investasilist()
{
    $data = $this->getInvestasi()->getContent();
    return view( view: 'investor.investasilist' )->with('data', json_decode($data, ASSOCIATIVE));
}
```

Gambar 5.29 Potongan Kode Front-end Melihat Investasi

Fungsi ini menerapkan microservice dengan potongan kode pada gambar 5.30.

```
public function getInvestasi()
{
    $id = Auth::user()->id;
    $data = Investasi::where ('user_investor_id', $id)->get();
    return response()->json(['data' => $data]);
}
```

Gambar 5.30 Potongan Kode Microservice Melihat Investasi

5.2.12 Melihat List Umkm

Fungsi melihat list ukm digunakan oleh investor apabila ingin melakukan investasi. Tampilan dari fungsi ini dapat dilihat pada gambar 5.31.

No	Nama	Email	Phone	Action
1	Geprekrers - UKM User	ukm@gmail.com	031111113	Detail
2	GeprukGeprek - UKM User	ukm1@gmail.com	034444444	Detail
3	Ukm3	ukm@gmail.com	8739812121	Detail

Gambar 5.31 Tampilan Front-end List Umkm

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.32.

```
public function listumkm()
{
    $data = $this->getUmkm()->getContent();
    return view( view: 'investor.listumkm')->with('data', json_decode($data, ASSOC true));
}
```

Gambar 5.32 Potongan Kode Front-end List Ukm

Fungsi ini menerapkan microservice dengan potongan kode seperti pada gambar 5.33.

```
public function getUmkm()
{
    $data = User::where('usertype_id', 2)->get();
    return response()->json(['data' => $data
    ]);
}
```

Gambar 5.33 Potongan KodeMicroservice List Ukm

5.2.13 Melihat detail Umkm

Melihat detail Umkm merupakan fungsi yang dapat melihat umkm lebih detail mengenai datanya agar dapat digunakan pertimbangan investor. Tampilan Melihat Detail Umkm dapat dilihat pada gambar 5.34.

Profil UMKM	
Name	Geprekers - UMK User
User Type	Umkm
Jenis UMKM	UMK Geprek
Jenis Usaha	Makanan cepat saji
Tahun Berdiri	2005
Alamat Usaha	Klampis ngasem, Surabaya
Telepon	0856789098
Jumlah Tenaga Kerja	18
Legalitas Usaha	HHK/090/JYIOL
Penjualan Per Bulan	100 item / hari
<div>Back</div>	

Gambar 5.34 Tampilan Front-end Detail Ukm

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.35.

```
public function ukmdetail($id)
{
    $data = $this->getUkmDetail()->getContent();
    return view( view: 'investor.ukmdetail' )->with('data', json_decode($data, assoc: true));
}
```

Gambar 5.35 Potongan Kode Front-end Detail Ukm

Fungsi ini menerapkan microservice dengan potongan kode seperti pada gambar 5.36.

```
public function getUkmDetail($id)
{
    $data = User::find($id);
    return response()->json(['data' => $data
    ]);
}
```

Gambar 5.36 Potongan Kode Microservice Detail Ukm

5.2.14 Melihat Ukm yang aktif coaching

Fungsi Melihat Ukm yang aktif Coaching digunakan oleh Coach sebelum melakukan chatting dengan umkm untuk mengetahui siapa saja umkm yang aktif. Tampilan fungsi ini dapat dilihat pada gambar 5.37.

Mentor UKM		
No	Permasalahan	Chat UKM
1	Omset belum banyak	Geprekers - UKM User
2	q	GeprukGeprek - UKM User

Gambar 5.37 Tampilan Front-end Ukm Aktif Coaching

Untuk mendapatkan tampilan tersebut, dibuat potongan kode seperti pada gambar 5.38.

```

public function history($id)
{
    if ($id == null) {
    } else {
        $data = $this->getListUkm()->getContent();
        return view('coach.history')->with('data', json_decode($data, assoc: true));
    }
}

```

Gambar 5.38 Potongan Kode Front-end Ukm Aktif Coaching

Fungsi ini menerapkan microservice dengan potongan kode seperti pada gambar 5.39.

```

public function getListUkm()
{
    $user_id = Auth::user()->id;
    $data = Coaching::where('user_coach_id', $user_id)->where('coachingstatus_id', 2)->get();
    return response()->json(['data' => $data]);
}

```

Gambar 5.39 Potongan Kode Microservice Ukm Aktif Coaching

5.3 Implementasi Test Case

Pada table menjelaskan implementasi pengujian yang dilakukan untuk menguji kebutuhan fungsional sistem apa sudah berjalan sesuai dengan use case.

Tabel 5.3 Skenario Uji Coba

No	Skenario	Prosedur Pengujian	Masukan	Ekspektasi Luaran
1	Normal	Membukan halaman login aplikasi Memasukkan email dan passwrod yang telah terdaftar	Email dan Password	Berhasil masuk ke halaman dashboard sesuai dengan role
2	Skenario Alternatif	Membukan halaman login aplikasi Memasukkan email dan password yang tidak terdaftar	Email dan password	User tidak berhasil masuk dan keluar pesan data tidak sesuai.

3	Normal	Malakukan penambahan data user oleh Admin dengan data yang belum ada	Data User	Berhasil menambahkan data
4	Alternatif	Malakukan penambahan data user oleh Admin dengan email yang sudah terdaftar	Data User	Peringatan eror data sudah ada.
5	Normal	Melakukan investasi dengan data lengkap	Data Investasi	Data berhasil tersimpan
6	Alternatif	Melakukan investasi dengan tidak data lengkap	Data Investasi	Muncul Peringatan memasukkan data

5.4 Implementasi REST API pada perangkat lain

Dalam implementasi Arsitektur Microservice, service tertentu yang telah dibuat dapat digunakan oleh aplikasi lain untuk mendapatkan data yang diinginkan. Dalam implementasi dilakukan uji coba terhadap service list ukm yang menampilkan ukm apa saja ada dalam aplikasi. Pada pengujian kali ini dilakukan untuk membuat aplikasi android yang dikembangkan dengan bahasa pemrograman Java. Kode program dalam menggunakan service dapat dilihat pada gambar 5.40.

```

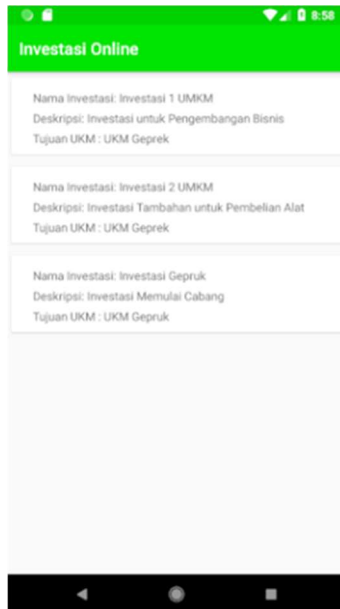
public class ApiClient {
    public static final String BASE_URL = "http://192.168.100.11/einkubator-api/";

    public static Retrofit retrofit=null;

    public static Retrofit getApiClient() {
        if (retrofit==null) {
            retrofit=new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}

```

Gambar 5.40 Penggunaan API pada Java Android



Gambar 5.41 Tampilan Hasil pada Android

Dari gambar 5.41 menjelaskan bahwa service pada aplikasi yang telah dibuat menggunakan microservice dapat di gunakan pada aplikasi lain.

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan diuraikan hasil dan pembahasan terhadap apa yang sudah dilakukan pada tugas akhir ini.

6.1 Arsitektur Microservice

Hasil pengujian service yang telah dibuat menggunakan tools Apache Benchmark memperlihatkan bahwa penggunaan microservice menghasilkan Request Time per Second sebesar 20 RPS terlihat pada gambar 6.1.

```
Concurrency Level:      100
Time taken for tests:    5.063 seconds
Complete requests:      104
Failed requests:         4
  (Connect: 0, Receive: 0, Length: 4, Exceptions: 0)
Non-2xx responses:      104
Total transferred:      165506 bytes
HTML transferred:       55664 bytes
Requests per second:    20.54 [#/sec] (mean)
Time per request:       4868.378 [ms] (mean)
Time per request:       48.684 [ms] (mean, across all concurrent requests)
Transfer rate:          31.92 [Kbytes/sec] received
```

Gambar 6.1 RPS Microservice

Sedangkan RPS dari arsitektur monolitik memiliki RPS sebesar 8.19 RPS terlihat pada gambar 6.2.

```
Concurrency Level:      100
Time taken for tests:    5.252 seconds
Complete requests:      43
Failed requests:        42
  (Connect: 0, Receive: 0, Length: 42, Exceptions: 0)
Non-2xx responses:      44
Total transferred:      28468878 bytes
HTML transferred:       28424264 bytes
Requests per second:    8.19 [#/sec] (mean)
Time per request:       12213.151 [ms] (mean)
Time per request:       122.132 [ms] (mean, across all concurrent requests)
Transfer rate:          5293.88 [Kbytes/sec] received
```

Gambar 6.2 RPS Monolitik

Dari hasil percobaan, dilakukan percobaan beberapa kali dengan mengubah variabel durasi percobaan dan jumlah service yang bisa dilakukan dalam satu waktu terlihat pada tabel 6.1 dengan t merupakan durasi waktu dan n merupakan jumlah service yang bisa dilakukan dalam satu waktu.

Tabel 6.1 Hasil Uji Coba Load Testing

No	Waktu Eksekusi (detik)	Tingkat Concurrency (Request)	RPS Mono (#/s)	RPS Micro (#/s)	Selisih (#/s)
1	10	10	2.97	7.42	4.45
2	50	10	3.01	7.50	4.49
3	100	10	2.91	6.78	3.87
4	150	10	2.99	6.39	3.40
5	200	10	2.95	6.16	3.21
6	250	10	2.91	7.96	5.05
7	300	10	2.88	7.69	4.81

Load testing digunakan untuk mengukur bagaimana suatu web dapat diakses dengan parameter durasi eksekusi dan jumlah service yang dapat diakses secara bersamaan dalam satu waktu. Dari load testing didapatkan hasil berupa RPS atau Request Per Second, yang berarti jumlah permintaan yang bisa diterima oleh web server dalam satu waktu. Dalam setiap pengujian yang dilakukan, di dapatkan bahwa RPS dari microservice lebih tinggi daripada arsitektur monolitik.

6.2 Stress Testing

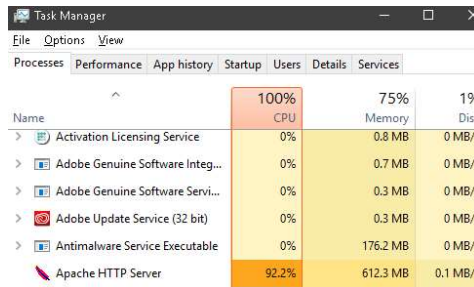
Stress Testing dilakukan dengan menggunakan Apache Benchmark dengan parameter berupa jumlah service yang akan di minta dan jumlah service yang diminta secara bersamaan. Pada table 6.2 merupakan hasil uji coba pada tingkat stress dari arsitektur monolitik dan microservice .

Tabel 6.2 Hasil Uji Coba Stress Testing

No	Jumlah Permintaan (Request)	Tingkat Concurrency (Request)	RPS Mono (#/s)	RPS Micro (#/s)
1	10000	200	21.80	10.25
2	5000	200	19.19	9.30

3	2000	200	16.71	8.67
4	1000	200	15.92	8.63

Stress Testing digunakan untuk mengukur bagaimana suatu web dapat bertahan atau masih dapat melayani permintaan apabila terjadi tingkat permintaan yang tinggi dan jumlah permintaan dalam satu waktu juga tinggi. Dalam percobaan kali ini dilakukan dengan menggunakan parameter n dan c , dimana n merupakan jumlah service yang bisa diterima oleh web server dan c merupakan jumlah service yang bisa diterima oleh web server dalam satu waktu. Hasil dari percobaan menunjukkan bahwa arsitektur microservice memiliki tingkat RPS yang tinggi daripada arsitektur monolitik.



Gambar 6.3 Performa CPU dan RAM

Pada waktu service yang bisa diterima mencapai 10000, penggunaan CPU sudah mencapai 100% dan RAM sebesar 75%.

6.3 Penggunaan service pada Android

Dalam implementasi telah dibuat service yang dapat digunakan pada perangkat lain. Dalam penggunaan pada perangkat lain dalam hal ini Android, dilakukan pengujian perhitungan terhadap *Response Time* dari service tersebut seperti pada gambar 6.4.

```

public void onResponse(Call<UkmList> call, Response<UkmList> response) {
    ukmList = response.body().getData();
    long tx = response.raw().sentRequestAtMillis();
    long rx = response.raw().receivedResponseAtMillis();
    System.out.println("response time : +(rx - tx)+" ms");
    List<String> listSpinner = new ArrayList<>();
    for(int i=0;i<ukmList.size();i++){
        listSpinner.add(ukmList.get(i).getName());
    }
}

```

Gambar 6.4 Kode Menghitung Response Time Service

Pada implementasi *testing* di Android, fungsi *sentRequestAtMillis* dan *receivedResponseAtMillis* pada package Retrofit digunakan untuk menghitung Response Time dari sistem. Dalam beberapa uji coba yang dilakukan untuk menghitung response time dari service didapatkan hasil seperti pada gambar 6.5.

```

4823-4823/com.brodan.android.inkubatoronline I/System.out: response time : 321 ms
4823-4823/com.brodan.android.inkubatoronline I/System.out: response time : 373 ms
4823-4823/com.brodan.android.inkubatoronline I/System.out: response time : 283 ms

```

Gambar 6.5 Hasil Uji Coba Response Time

Dari hasil yang didapat, dapat diketahui bahwa rata-rata response time dari service untuk melihat data ukm pada perangkat lain dalam kasus kali ini pada android adalah sebesar 325 ms.

BAB VII

KESIMPULAN DAN SARAN

Bab kesimpulan dan saran membahas mengenai kesimpulan proses penelitian yang telah dilakukan dan saran yang diusulkan untuk penelitian serupa di masa mendatang.

7.1 Kesimpulan

Dari proses yang telah dilalui dalam pengerjaan tugas akhir ini, didapatkan kesimpulan sebagai berikut.

1. Aplikasi E-Inkubator menggunakan arsitektur microservice memiliki tingkat RPS yang lebih tinggi daripada aplikasi yang menggunakan arsitektur monolitik.
2. Layanan/ service yang diimplementasikan dalam aplikasi ini dapat digunakan oleh aplikasi lain karena service yang dibuat menggunakan metode komunikasi data Rest API.

7.2 Saran

Dalam proses pengerjaan tugas akhir ini, terdapat beberapa saran untuk penelitian mendatang sebagai berikut.

1. Penelitian selanjutnya dapat menggunakan whitebox testing agar tiap unit yang dites dapat diuji secara mendalam ke bagian kodenya.
2. Dalam aplikasi E-Inkubator, terdapat beberapa fungsi yang masih menggunakan arsitektur monolitik/ tidak semua fungsi pada aplikasi diimplementasikan arsitektur microservice, sehingga kedepan dapat diimplementasikan ke semua fungsi seperti lelang.
3. Dalam melakukan testing arsitektur microservice, dilakukan dengan melihat Request Per Second.

Kedepannya, lebih baik dapat dilakukan metode lain untuk memperlihatkan pengaruh dari penerapan arsitektur microservice.

DAFTAR PUSTAKA

- [1] Mario Villamizar et al., "Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud," IEEE, pp. 583-590, 2015.
- [2] Chris Richardson. 2017. Microservice.io. <<https://microservice.io/patterns/monolithic.html>>
- [3] S Kustin Ayuwuragil. 2017. CNN Indonesia. <<https://www.cnnindonesia.com/ekonomi/20171115161037-78-255819/kemenkop-ukm-379-juta-umkm-sudah-go-online>>
- [4] Joe Stubbs, Walter Moreira, and Rion Dooley, "Distributed Systems of Microservice Using Docker and Serfnode," Budapest, 2015.
- [5] G. Kecskemeti, A. Kertesz, and A. C. Marosi, "Towards a Methodology to form Microservice from Monolithic Ones," no. 644179.
- [6] Sourabh Sharma, "Mastering Microservice with Java," in Mastering Microservice with Java. Birmingham:PACKT publishing, 2016.
- [7] Rajesh RV, "Spring Microservice," in Spring Microservice. Birmingham: PACKT publishing, 2016.
- [8] Rama Rahmanda, "Rancang Bangun Aplikasi berbasis Microservice untuk Klasifikasi Sentimen. Studi Kasus: PT. Yesboss Group Indonesia (Kata.ai)" Institut Teknologi Sepuluh Nopember, 2017.
- [9] Dedy Puji Jayanto, "Rancang Bangun Back-end Aplikasi "SIAP: Sistem Informasi Aspirasi dan Pengaduan Masyarakat" serbasis Web/Sms dengan menggunakan Metode Microservice Spring Boot" Institut Teknologi Sepuluh Nopember, 2017.
- [10] Jarman, Peter. 2017. Microservice – A New Application

Paradigm. Infosys.

- [11] Boyke Dian Triwahyudhi. April 2016. indosystem. <<https://indosystem.com/blog/microservice-konsep-dan-implementasi>>.
- [12] Bean, M. (2015). *Laravel 5 Essentials*. Birmingham: Packt Publishing Ltd.
- [13] Undang-undang Nomor 20 Tahun 2008. Tentang Usaha Mikro, Kecil dan Menengah.
- [14] Susi Susilowati, "Pengembangan Sistem Informasi Manajemen Zakat, Infaq, Shadaqoh, Waqaf dan Hibah menggunakan Metode Waterfall," Paradigma, 2017
- [15] Manish Kumar, Santos Kumar Singh, and Dwivedi, "A Comparative Study of Black Box Testing and White Box Testing Techniques," *International Journal of Advance Research in Computer Science and Management Studies*, 2015.
- [16] Syed Roohullah Jan, Syed Tauhid Ullah Shah, Zia Ullah Johar, Yasin Shah, and Fazlullah Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *IJSRSET*, 2016.
- [17] Alessandro Orso and Gregg Rothermel, "Software testing: a research travelogue (2000–2014)," in *Proceedings of the on Future of Software Engineering*, New York, 2014.
- [18] H. Suryotrisongko, "Arsitektur Microservice untuk Resiliensi Sistem Informasi," vol. 6, no. 2, pp. 235–250, 2017.

LAMPIRAN A

A. Hasil Load Testing

No	Waktu Eksekusi (detik)	Tingkat Concurrency (Request)	RPS Mono (#/s)	RPS Micro (#/s)	Selisih (#/s)
1	10	10	2.97	7.42	4.45
2	50	10	3.01	7.50	4.49
3	100	10	2.91	6.78	3.87
4	150	10	2.99	6.39	3.40
5	200	10	2.95	6.16	3.21
6	250	10	2.91	7.96	5.05
7	300	10	2.88	7.69	4.81
8	10	50	2.86	7.84	4.98
9	50	50	2.9	7.51	4.61
10	100	50	2.95	7.36	4.41
11	150	50	2.99	6.70	3.71
12	200	50	3	7.22	4.22
13	250	50	2.89	6.78	3.89
14	300	50	2.94	6.36	3.42
15	10	100	3.01	6.34	3.33
16	50	100	3.05	7.59	4.54
17	100	100	3.01	7.32	4.31
18	150	100	2.9	7.98	5.08
19	200	100	3.01	6.42	3.41
20	250	100	2.98	6.85	3.87
21	300	100	2.89	6.12	3.23
22	10	150	3	7.01	4.01
23	50	150	3.01	6.09	3.08
24	100	150	2.97	6.80	3.83
25	150	150	2.98	6.73	3.75
26	200	150	3.02	6.40	3.38
27	250	150	3.05	7.14	4.09
28	300	150	3.12	6.20	3.08
29	10	200	2.99	6.87	3.88
30	50	200	2.98	7.15	4.17
31	100	200	3.01	6.59	3.58
32	150	200	3.1	6.25	3.15
33	200	200	2.99	6.54	3.55
34	250	200	2.98	7.88	4.90
35	300	200	3.01	7.00	3.99
36	10	250	2.98	6.23	3.25
37	50	250	3	7.96	4.96

38	100	250	3.01	6.10	3.09
39	150	250	3.05	6.52	3.47
40	200	250	3	7.89	4.89
41	250	250	3.01	7.26	4.25
42	300	250	3.02	7.50	4.48
43	10	300	3.01	6.55	3.54
44	50	300	2.98	7.67	4.69
45	100	300	2.99	7.06	4.07
46	150	300	3.05	7.63	4.58
47	200	300	3.12	7.37	4.25
48	250	300	3.05	6.72	3.67
49	300	300	3.01	6.09	3.08
Rata-rata			2.99	6.97	3.98

BIODATA PENULIS



Wildan Azka Fillah, lahir di Lamongan 26 Oktober 1998 merupakan anak kedua dari dua bersaudara yang dibesarkan di desa kecil di daerah Lamongan. Penulis menempuh pendidikan dasar di SD N Sudangan, kemudian melanjutkan studi ke MTs N Lamongan, lalu masuk ke program IPA Akselerasi MAN Lamongan. Lulus dari MAN Lamongan pada tahun 2015 program IPA Akselerasi, penulis melanjutkan jenjang pendidikan di Departemen

Sistem Informasi Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan NRP 5215100077. Selama kuliah, selain mengikuti proses pembelajaran, penulis juga mengikuti beberapa kegiatan kemahasiswaan di kampus dan beberapa kali mengikuti kegiatan yang mengasah keprofesian penulis. Ketertarikan penulis akan teknologi informasi dan komunikasi serta bisnis membuat penulis ikut dalam suatu organisasi yaitu Koperasi Mahasiswa. Pada tahun pertama, penulis mengikuti beberapa kegiatan kemahasiswaan baik dari departemen baik itu dari Himpunan Mahasiswa Sistem Informasi (HMSI) maupun Kajian Islam Sistem Informasi (KISI). Pada tahun kedua, penulis berfokus pada Kopma dan KISI. Selain itu, penulis juga mulai aktif dalam mengikuti perlombaan keprofesionalan seperti PKM, Gemastik, dan Comptech. Topik yang diambil oleh penulis merupakan bidang minat dari Laboratorium Infrastruktur dan Keamanan Teknologi Informasi. Penulis dapat dihubungi melalui email: wildanazkafillah@gmail.com atau nomor telepon +6281515412541.